A3 Clarifications

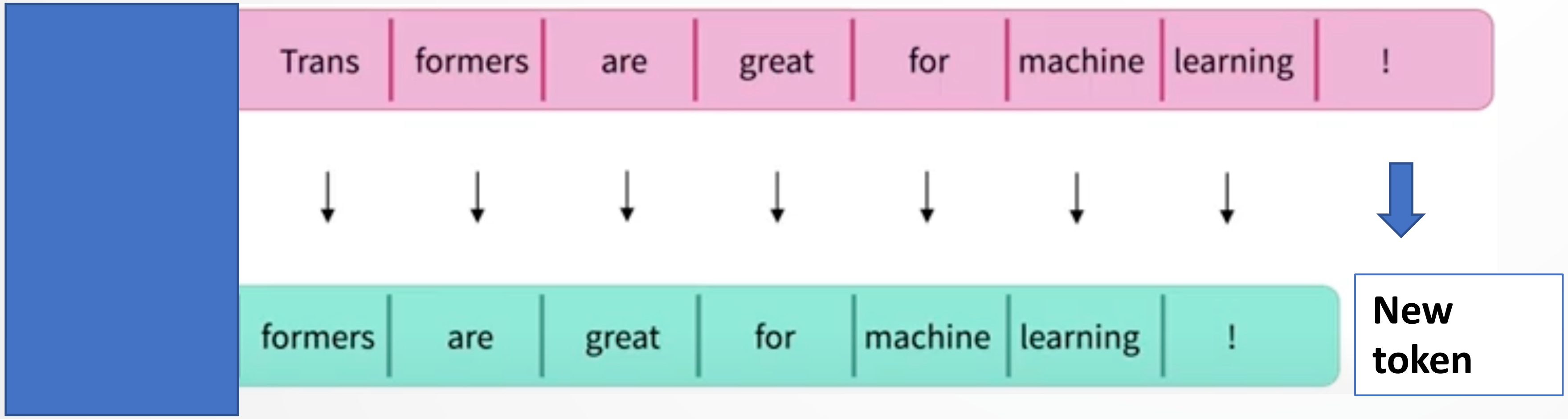
- Where to apply softmax at the classifier head?
  - After the classifier head.

- Model saving / loading:
  - torch.save(model.state_dict(), f)
  - model.load_state_dict(state_dict)

- What is AutoModelForCausalLM?
  - It is still a sequential classification model, but it predicts the probability of the next token.
  - I.e., the classifier is Linear(D, vocab_size) instead of Linear(D, 2)

UNIVERSITY OF TORONTO

A3: AutoModelForCausalLM



Aside: A language model $P(w_t|w_{<t})$ can be used to generate a sequence.

Image source: AutoModelForCausalLM tutorial: https://huggingface.co/course/chapter7/6

Automatic Speech Recognition

CSC401/2511 – Natural Language Computing – Winter 2023

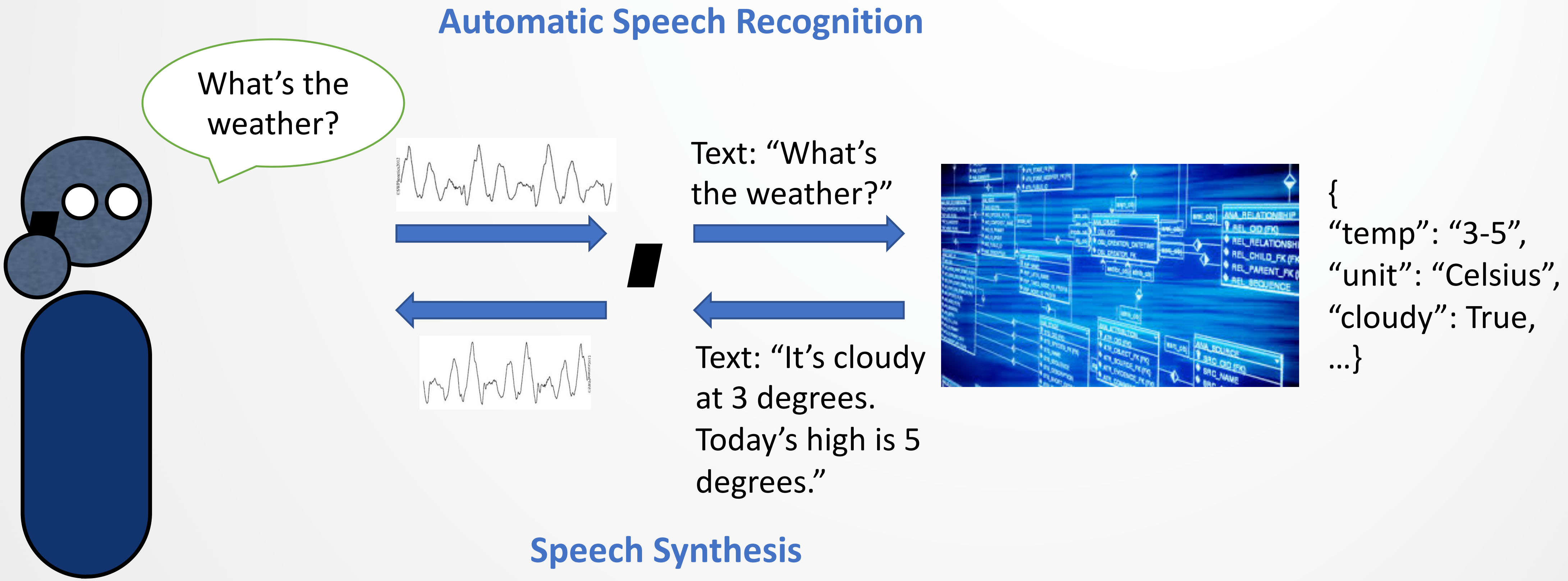
University of Toronto

Contents

- Today's lecture:
  - What is ASR?
  - A noisy-channel ASR model.
- Next lecture:
  - An end-to-end ASR model.
  - Evaluation

Applications of speech technology

A use case



UNIVERSITY OF TORONTO

What is ASR?

Automatic Speech Recognition (ASR) systems converts **speech** into **text**.

- Input: speech data X
- Output: text W
- Other names for the system: speech recognition, speech-to-text.

Previous lectures described texts.

Next slide: the formats of speech data X.

In 20 mins: what speech actually is.

Formats of speech data X

- Raw speech data are 1-d arrays of shape (T_1)
  - $T_1 = f \times t$
  - f is **sample rate** (e.g., 16kHz)
  - t is **time length**, in seconds.

- We can also **extract** speech features.
  - The speech features are 2-d arrays of shape (T_2, D)
  - $T_2 \ll T_1$
  - D: Number of features
  - Lecture: spectrum. Tutorial: MFCC feature.

A simplified system

Two assumptions:

[A1] Each word W has one and only one sound X_W.

[A2] Each speech sample X contains exactly one word.

Then ASR can be addressed by **maximum similarity search**:

$$W = \max_{w} \operatorname{sim}\langle X, X_w \rangle$$

UNIVERSITY OF TORONTO

Let’s relax one assumption

[A1’] Each word w might have different sounds X.

We can use an **acoustic model** $P(X|w)$ to model.

[A2] Each speech sample X still contains exactly one word.

- This ASR system then becomes:

$$\mathrm{W} = \max_{\mathrm{w}} P(w|X)$$

UNIVERSITY OF TORONTO

Recall: Bayes' Theorem

$$P(w|X) = \frac{P(X|w)P(w)}{P(X)}$$

- $P(w)$: prior probability — Language model
- $P(X|w)$: likelihood — Acoustic model
- $P(w|X)$: posterior probability

Putting them together

$$W = \max_{w} P(w|X) = \max_{w} \frac{P(X|w)P(w)}{P(X)}$$

Since $P(X)$ is constant wrt w, it doesn't matter here.

$$W = \max_{w} P(X|w)P(w)$$

Noisy channel ASR

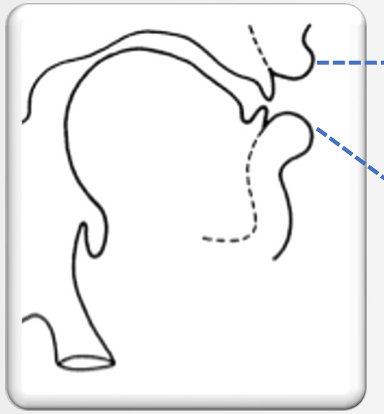
- Recall the assumption [A1']: Each word has multiple sounds.
- Consider speaking as a communication **channel**:
  - Pitch
  - Tone
  - Speed
  - … there are many factors that make this channel noisy.
- W to X goes through a **noisy channel**.
- The ASR model recognizes speech from this noisy channel.
- This is therefore a **noisy channel ASR model**.

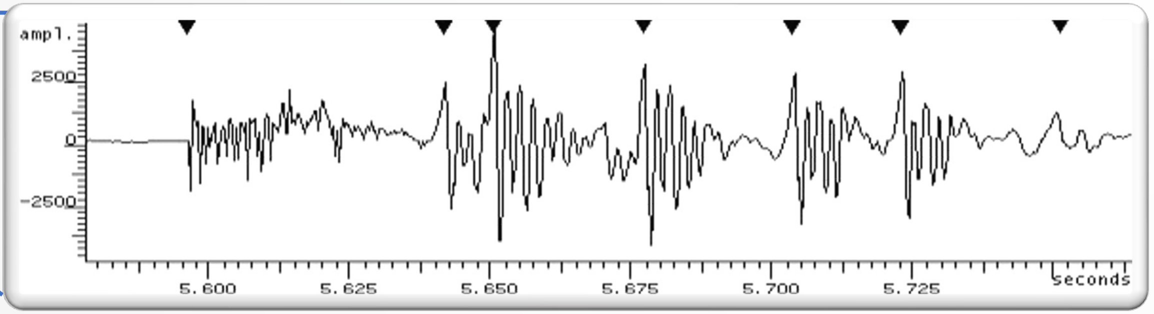
UNIVERSITY OF TORONTO

Historical notes on noisy channels

- Noisy channel are very popular.
- In machine translation, we also discussed noisy channel models.
- Since 2010+, these problems are frequently addressed by **sequence-to-sequence** models.
- Since 2020+, these problems are frequently addressed by Transformer-based models.

More on the speech





Is one-dimensional X the best input for our ASR systems?

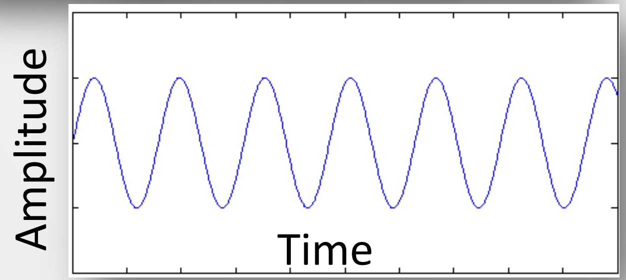
Speech constitutes of sounds

- **Sound** is a time-variant pressure wave created by a **vibration**.
  - Air particles **hit** each other, setting others in motion.
    - High pressure ≡ **compressions** in the air (C).
    - Low pressure ≡ **rarefactions** within the air (R).

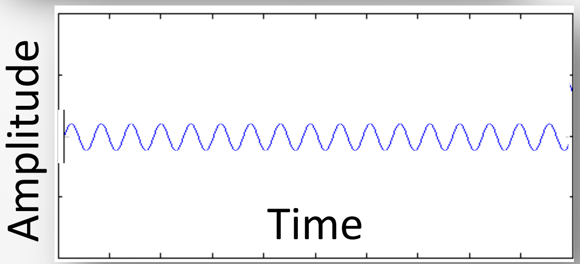
Amplitude and frequency of sound

- A single **tone** is a sinusoidal function of pressure and time.
  - **Amplitude**: *n.* The degree of the displacement in the air. This is similar to 'loudness'. Often measured in **Decibels (dB)**.
  - **Frequency**: *n.* The number of cycles within a unit of time. e.g., **1 Hertz (Hz) = 1 oscillation/second**

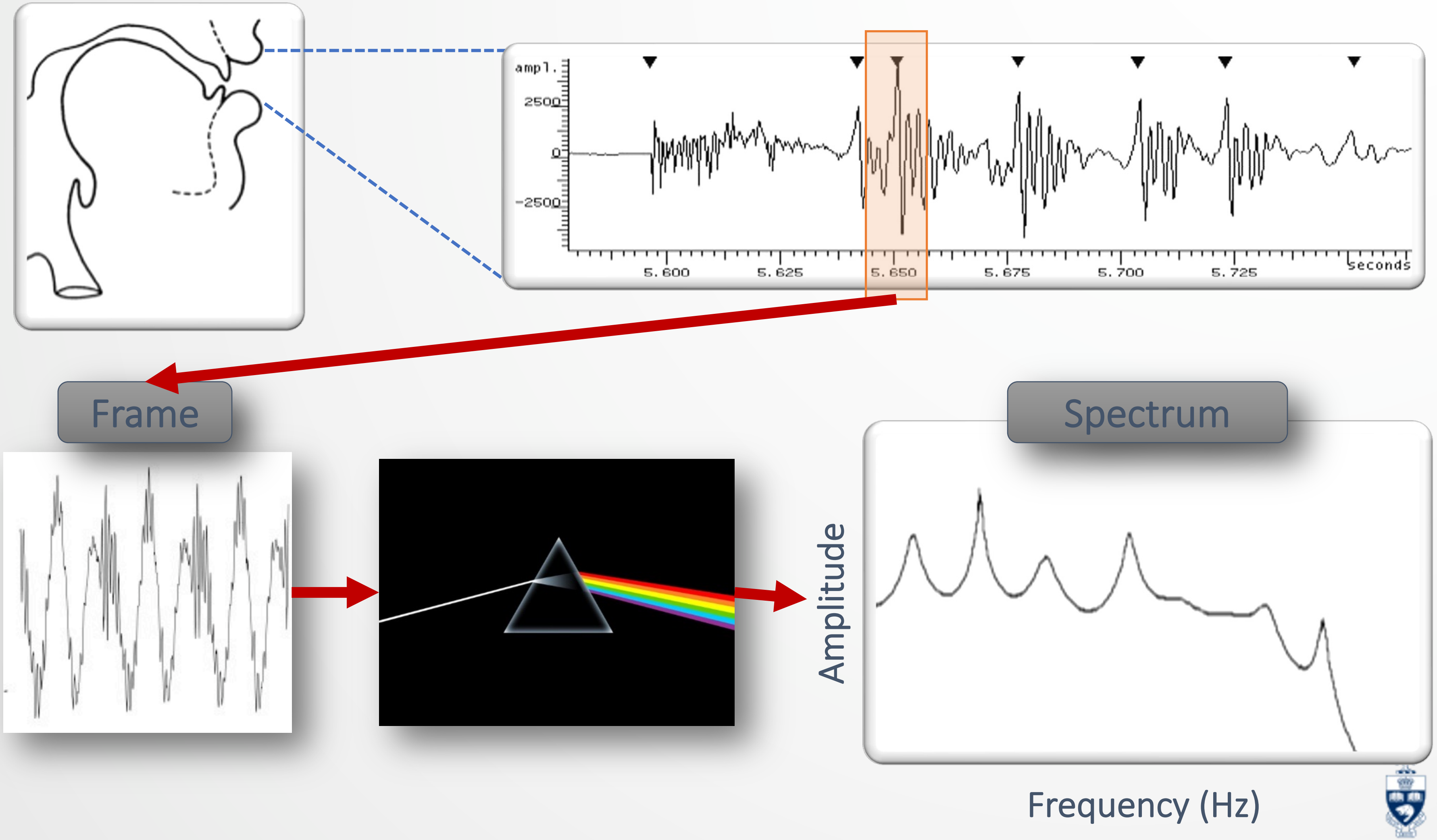
Lower frequency, higher amplitude



Higher frequency, lower amplitude

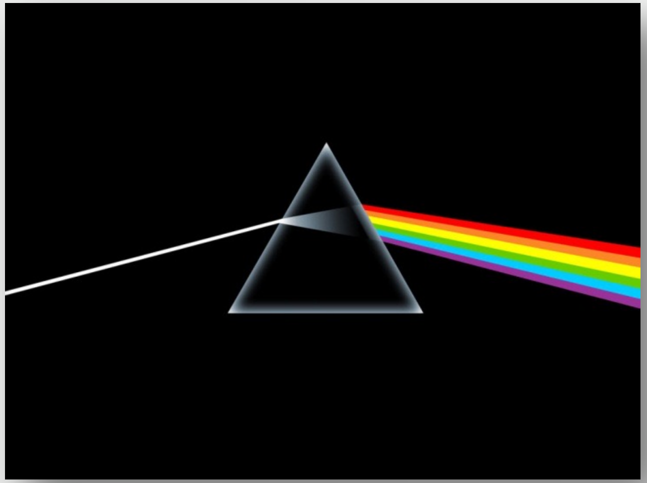


Extract features from X



Spectrum captures the amplitudes at different frequency bands.

Aside: Fourier Transform



- **Input**: Continuous signal $x(t)$.
- **Output**: Spectrum $X(F)$

$$X(F) = \int_{-\infty}^{\infty} x(t)e^{-i2\pi Ft}\,dt$$

- It's **invertible**, i.e., $x(t) = \int_{-\infty}^{\infty} X(F)e^{i2\pi Ft}\,dF$.
- It's **linear**, i.e., for $a, b \in \mathbb{C}$,
  if $h(t) = ax(t) + by(t)$
  then $H(F) = aX(F) + bY(F)$

It needs **continuous** input $x(t)$…
Use **Discrete Fourier Transform**.


Fun fact: Fourier instructed Champollion.

UNIVERSITY OF TORONTO

Aside: implementing $\boldsymbol{P(X|W)}$

$P(X|W)$ can be implemented by:

- A nonparametric model based on training data.
- A Gaussian model.
- A neural network predicting P given X and W.
- … (whichever works well)

Lecture review questions

By the end of this lecture, you should be able to answer:

- What is ASR?
- What is speech?
- Describe some speech features:
  - Amplitude, frequency, spectrum
- Describe a noisy channel model for ASR.
  - What are its assumptions?
  - What are its inputs and outputs?

Anonymous feedback form: https://forms.gle/W3i6AHaE4uRx2FAJA

Limitations of last lecture's models

- We need a lot of data to model $P(X|W)$
- The $\max_{w}(\cdot)$ step takes a lot of time.
  - Nonstandard spelling increases the data requirements.

- The Bitter Lesson by Richard Sutton:
  - It's better to rely on DNN to learn by itself.

- This lecture: let's look at an end-to-end model, LAS.

Listen, Attend, Spell

- This is an **end-to-end** model.
- Many papers claimed their methods are end-to-end.
- Our definition for end-to-end is (loosely):
  - You only train one model.
  - Features (or sound waves) in, texts out.
  - There is no assembling of components.
- Sequence-to-sequence models are typically end-to-end models.

A schematic architecture

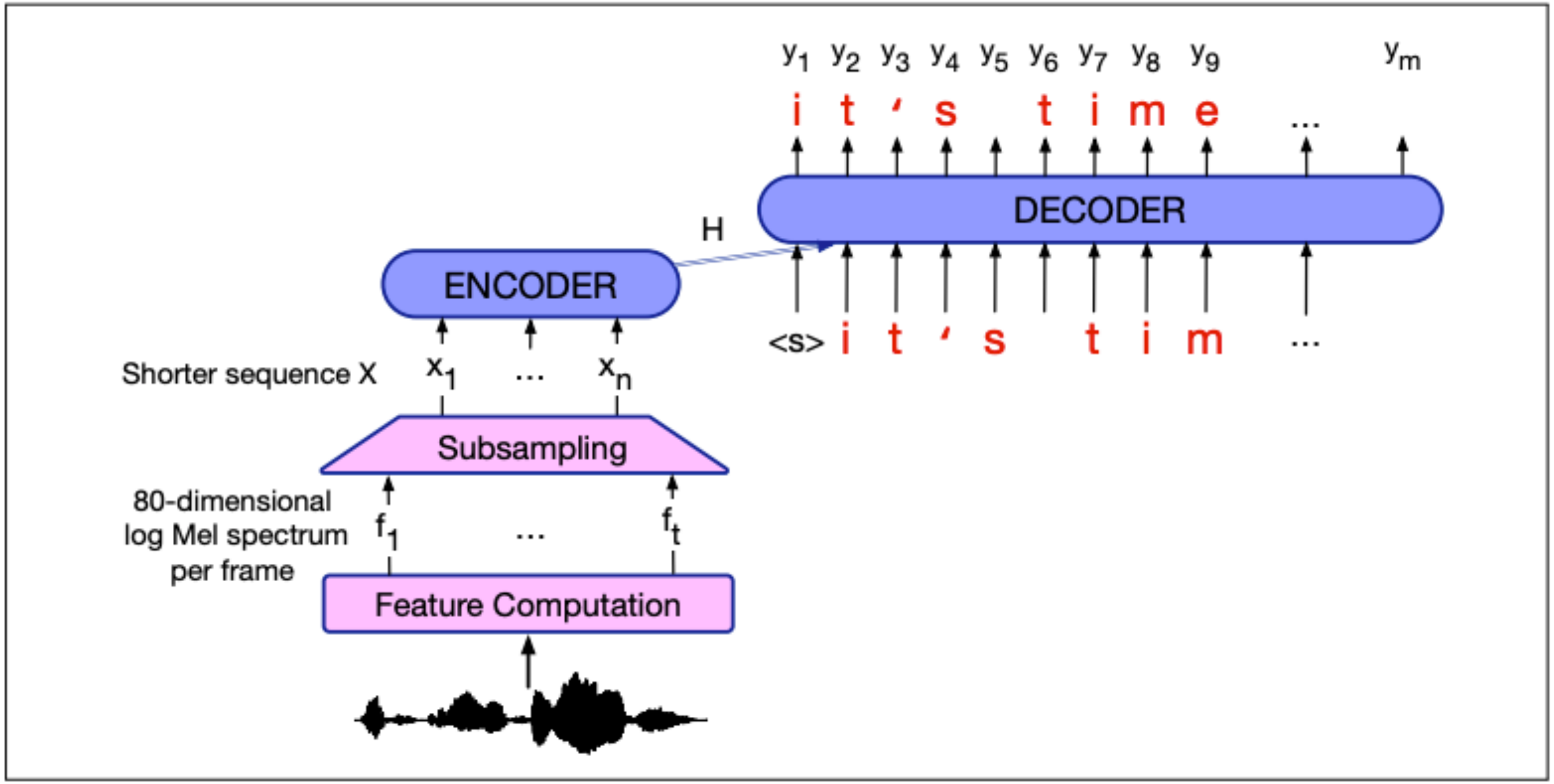


Figure 26.6 Schematic architecture for an encoder-decoder speech recognizer.

Figure source: Speech and Language Processing, Jurafsky & Martin, 3rd Edition

Model structure

- Encoder (“listener”): a pyramidial bidirectional LSTM.
- Decoder: RNN with attentions.
- Learning: teacher forcing.

$$\max_{\theta} \sum_{i} \log P(y_i | x, y^*_{<i}; \theta)$$

- x: sound features
- y^*: ground truth
- i: index of the word
- θ: model parameters

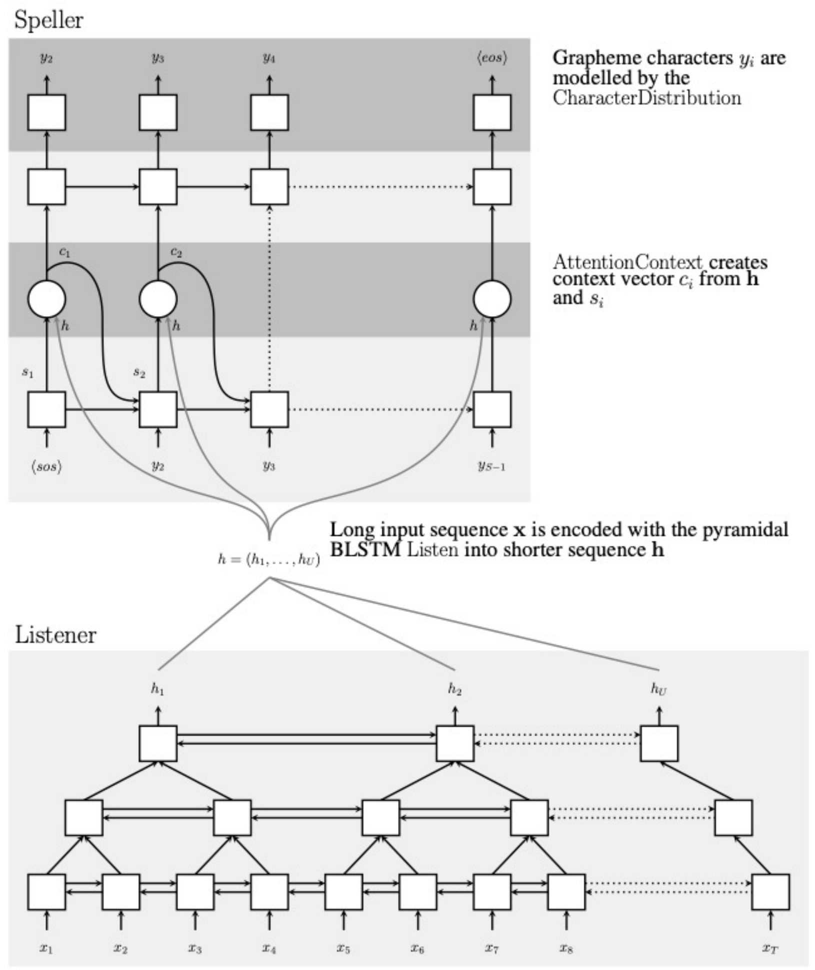


Figure 1: Listen, Attend and Spell (LAS) model: the listener is a pyramidal BLSTM encoding our input sequence **x** into high level features **h**, the speller is an attention-based decoder generating the **y** characters from **h**.

Model performance

- We'll talk about WER ("word error rates") later.

Table 1: WER comparison on the clean and noisy Google voice search task. The CLDNN-HMM system is the state-of-the-art system, the Listen, Attend and Spell (LAS) models are decoded with a beam size of 32. Language Model (LM) rescoring was applied to our beams, and a sampling trick was applied to bridge the gap between training and inference.

Model	**Clean WER**	**Noisy WER**
CLDNN-HMM [20]	8.0	8.9
LAS	16.2	19.0
LAS + LM Rescoring	12.6	14.7
LAS + Sampling	14.1	16.5
LAS + Sampling + LM Rescoring	10.3	12.0

Source: Listen, Attend and Spell https://arxiv.org/pdf/1508.01211.pdf

Aside: SOTAs in ASR

• LAS achieved SOTA in 2018, with new data and engineering.

Exp-ID	Model	WER
E7	WPM + MHA + Sync + SS + LS + MWER	5.8
E8	+ LM	**5.6**

Table 3: In second pass rescoring, the log-linear combination with a larger LM results in a 0.2% WER improvement.

Exp-ID	Model	VS/D	1st pass Model Size
E8	Proposed	**5.6/4.1**	**0.4 GB**
E9	Conventional LFR system	6.7/5.0	0.1 GB (AM) + 2.2 GB (PM) + 4.9 GB (LM) = 7.2GB

Table 5: Resulting WER on voice search (VS)/dictation (D). The improved LAS outperforms the conventional LFR system while being more compact. Both models use second-pass rescoring.

State-of-the-art Speech Recognition with S2S models. Chiu et al., (ICASSP 2018)

Aside: Recent topics in ASR

- Robustness towards types of noise
  - Differences in recording device.
  - Multiple speakers (where you need to do **speaker diarisation** too).
  - Background noise.
  - Dialects.
  - Slurred speech (e.g., when you speak faster than you think)
  - **Code switching** (-> this is *very* hard)
- VoicePrivacy: protect user identity without sacrificing the ASR performance.
  - To protect user identity, you add some noise.
  - But larger noise reduces the ASR performance!

Evaluating ASR System Performance

- if somebody said
  Reference: *how to recognize speech*

- but an ASR system reports
  Hypothesis: *how to wreck a nice beach*

- how do we quantify the error?

How to not measure ASR performance

- One measure is **word accuracy**: $\frac{N_{correct\_words}}{N_{reference\_words}}$
  - E.g., $2/4$, above
  - This runs into problems similar to those we saw with SMT.
    - E.g., the hypothesis '*how to recognize speech boing boing boing boing boing*' has 100% accuracy by this measure.
    - Normalizing by *#HypothesisWords* also has problems…

UNIVERSITY OF TORONTO

Word-error rate (WER)

- **Word-error rate (WER)** counts different **kinds** of errors that can be made by ASR at the word-level:

  - **Substitution error**: One word being mistook for another
    e.g., '***shift***' given '*ship*'
  - **Deletion error**: An input word that is 'skipped'
    e.g. '*I Torgo*' given '*I* ***am*** *Torgo*'
  - **Insertion error**: A 'hallucinated' word that was not in the input.
    e.g., '*This* ***Norwegian*** *parrot is no more*' given '*This parrot is no more*'

Word-error rate (WER)

- Putting them together:

$$WER = 100 \times \frac{N_{sub} + N_{ins} + N_{del}}{N_{words\ in\ reference}}$$

- Sometimes the types of errors can be weighed.
  - Because a substitution error is essentially a deletion followed by an insertion.
  - E.g., A substitution error weighs 5x as much as insertion or deletion error.

Computing WER

- $N = N_{sub} + N_{ins} + N_{del}$ can be computed by dynamic programming.
  - N is also called **edit distance**, or **Levenshtein distance**.
  - N is the **minimum** number of $N_{sub} + N_{ins} + N_{del}$ to edit the hypothesis H into the reference R.

- A **dynamic programming** algorithm contains:
  - An induction step, converting a “big problem” to a “smaller problem”.
  - An assumption that holds before and after the induction step.
  - A “base case”, where you know the answer easily.

UNIVERSITY OF TORONTO

Computing WER: induction

- Big problem and the assumption:

$N^{(n,m)}$ is the edit distance between the reference $R_{1..n}$ and the hypothesis $H_{1..m}$.

- Smaller problem and the assumption:

$N^{(n-1,m-1)}$ is the edit distance between the reference $R_{1..n-1}$ and the hypothesis $H_{1..m-1}$.

Computing WER: induction

- Given $N^{(n-1,m-1)}$ for the $R_{1..n-1}$, $H_{1..m-1}$ problem, how to compute $N^{(n,m)}$?
  - If $R_n == H_m$: $N^{(n,m)} = N^{(n-1,m-1)}$
  - Else: $N^{(n,m)} = 1 + \min\{N^{(n-1,m-1)}, N^{(n,m-1)}, N^{(n-1,m)}\}$

Substitution Deletion Insertion

Computing WER: initialization

- $N^{(0,m)} = m$
  - All of the hypotheses are hallucinated.
- $N^{(n,0)} = n$
  - All of the references are deleted.

Computing WER: implementation

		hypothesis							
		<s>	how	to	wreck	a	nice	beach	</s>
Reference	<s>	0	1	2	3	4	5	6	7
	how	1							
	to	2							
	recognize	3							
	speech	4							
	</s>	5							

UNIVERSITY OF TORONTO

Computing WER: implementation

		hypothesis							
		<s>	how	to	wreck	a	nice	beach	</s>
Reference	<s>	0	1	2	3	4	5	6	7
	how	1	0						
	to	2							
	recognize	3							
	speech	4							
	</s>	5							

UNIVERSITY OF TORONTO

Computing WER: implementation

		hypothesis							
		<s>	how	to	wreck	a	nice	beach	</s>
Reference	<s>	0	1	2	3	4	5	6	7
	how	1	0	1	2	3	4	5	6
	to	2							
	recognize	3							
	speech	4							
	</s>	5							

Computing WER: implementation

		hypothesis							
		<s>	how	to	wreck	a	nice	beach	</s>
Reference	<s>	0	1	2	3	4	5	6	7
	how	1	0	1	2	3	4	5	6
	to	2	1	0	1	2	3	4	5
	recognize	3	2	1	1	2	3	4	5
	speech	4							
	</s>	5							

Computing WER: implementation

		hypothesis							
		<s>	how	to	wreck	a	nice	beach	</s>
Reference	<s>	0	1	2	3	4	5	6	7
	how	1	0	1	2	3	4	5	6
	to	2	1	0	1	2	3	4	5
	recognize	3	2	1	1	2	3	4	5
	speech	4	3	2	2	2	3	4	5
	</s>	5	4	3	3	3	3	4	4

Lecture review questions

By the end of this lecture, you should be able to:

- Describe the *Listen, Attend, Spell* ASR model.
- Describe how to evaluate an ASR system.
  - What is word-error rate?
  - What is substitution / insertion / deletion error?
  - What are the initialization and induction steps to compute WER?
- (Optional) Do Leetcode Q72 “Edit Distance”.

Anonymous feedback form: https://forms.gle/W3i6AHaE4uRx2FAJA

UNIVERSITY OF TORONTO