Conversational UIs

Spoken Language Processing

Topic 3: Speech Recognition

Dr. Dong Minghui

Institute for Infocomm Research

Agency for Science, Technology and Research (A-Star)

Email: mhdong@i2r.a-star.edu.sg

PART 1. INTRODUCTION

What is Speech Recognition

- Automatic Speech Recognition (ASR) or Speech-to-Text is a process to automatically convert speech into text
- A natural way to input information to computer.
- The first step to understand speech. No understanding of the meaning yet.
- Convert digital signal (a sequence of continuous values) into text (discrete symbol representations)

Difficulties

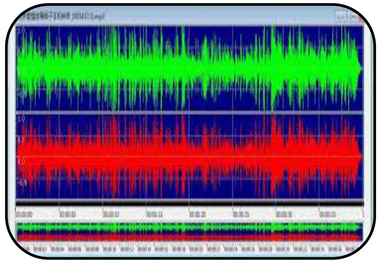
Device and Channel

Background noise

Speaker differences

Similar pronunciations

Accent, dialect

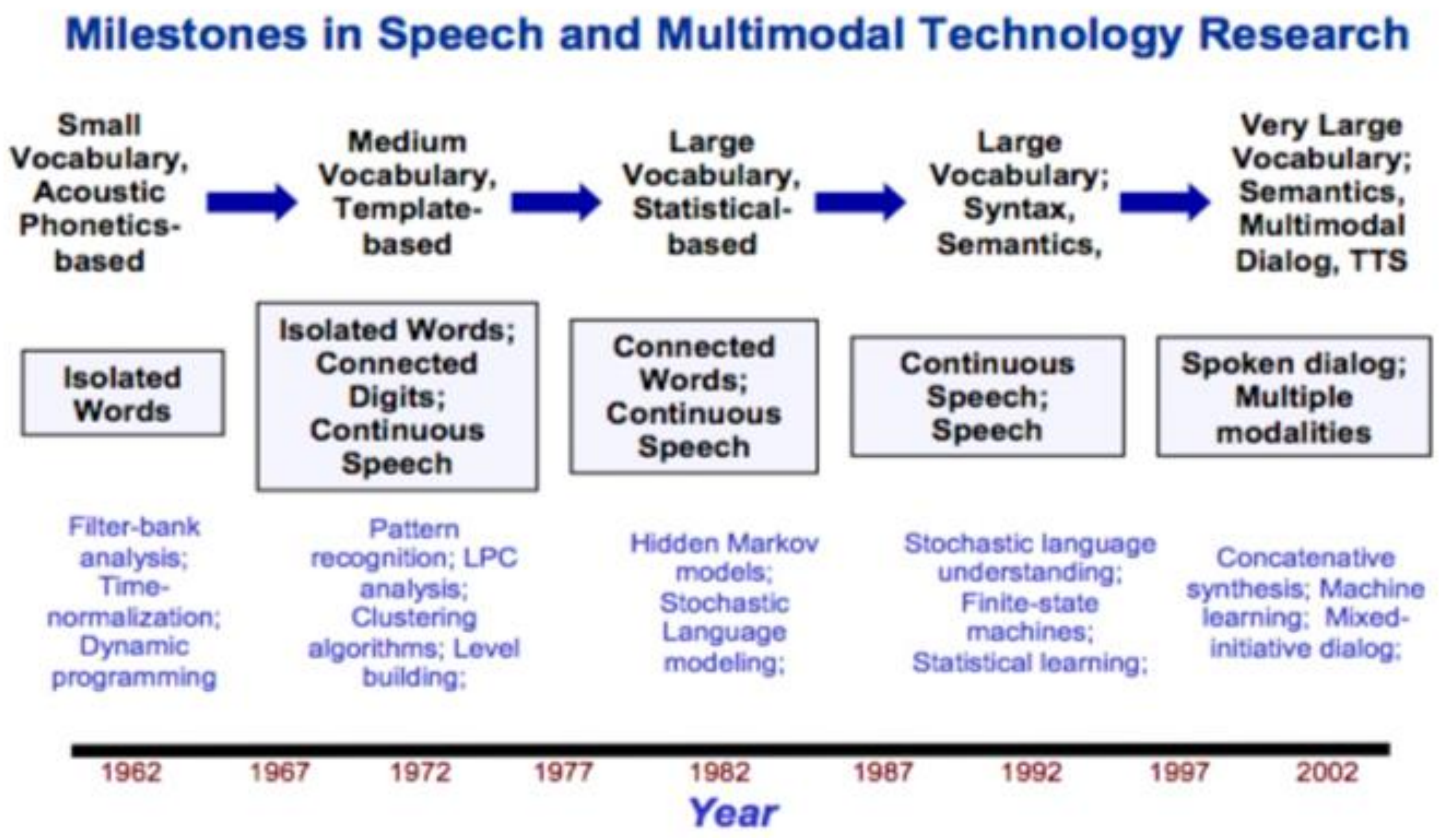


Pronunciation Variations in continuous speech

Prosody differences

Words with multiple pronunciations

ASR in early years



Milestones in Speech and Multimodal Technology Research

Small Vocabulary, Acoustic Phonetics-based → Medium Vocabulary, Template-based → Large Vocabulary, Statistical-based → Large Vocabulary; Syntax, Semantics, → Very Large Vocabulary; Semantics, Multimodal Dialog, TTS

Isolated Words

Isolated Words; Connected Digits; Continuous Speech

Connected Words; Continuous Speech

Continuous Speech; Speech

Spoken dialog; Multiple modalities

Filter-bank analysis; Time-normalization; Dynamic programming

Pattern recognition; LPC analysis; Clustering algorithms; Level building;

Hidden Markov models; Stochastic Language modeling;

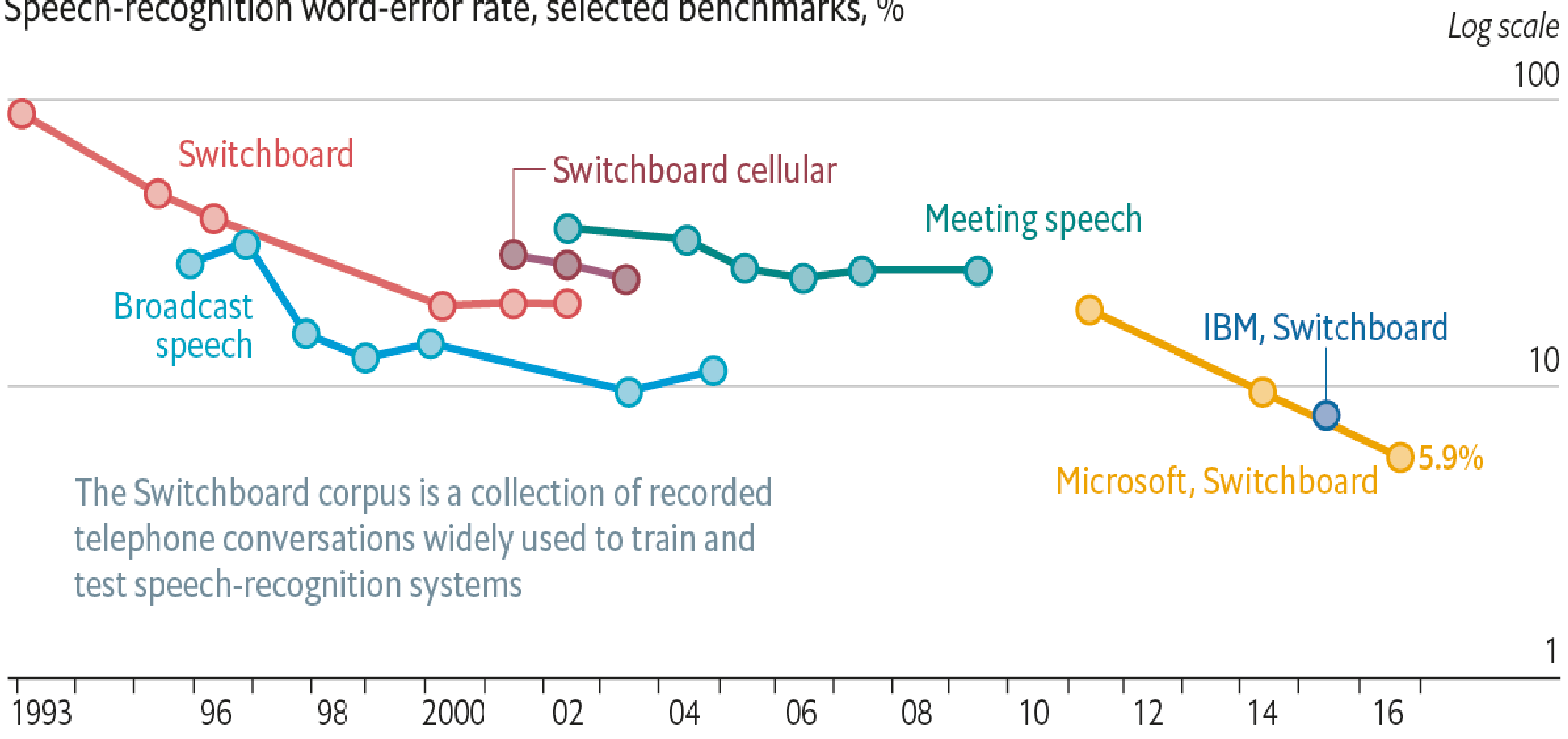
Stochastic language understanding; Finite-state machines; Statistical learning;

Concatenative synthesis; Machine learning; Mixed-initiative dialog;

ASR in recent years

Loud and clear

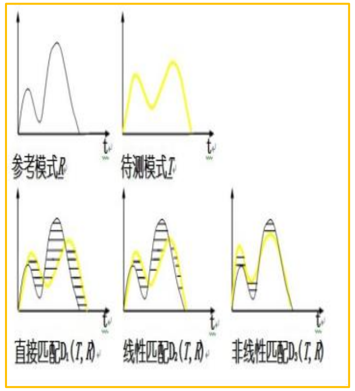
Speech-recognition word-error rate, selected benchmarks, %



The Switchboard corpus is a collection of recorded telephone conversations widely used to train and test speech-recognition systems

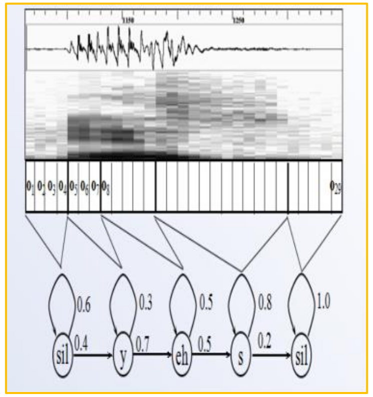
Sources: Microsoft; research papers

Evolution of ASR methods



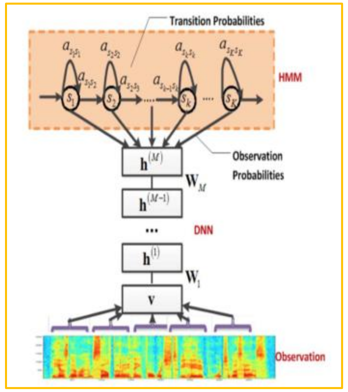
Pattern matching

- Since 1952
- Isolated words



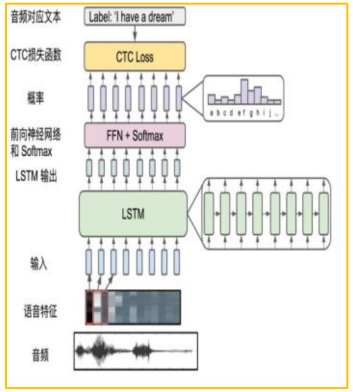
HMM models

- Since 1980
- GMM-HMM
- Continuous speech
- Limited accuracy



Deep learning

- Since 2009
- DNN-HMM
- Huge language model
- Natural speech

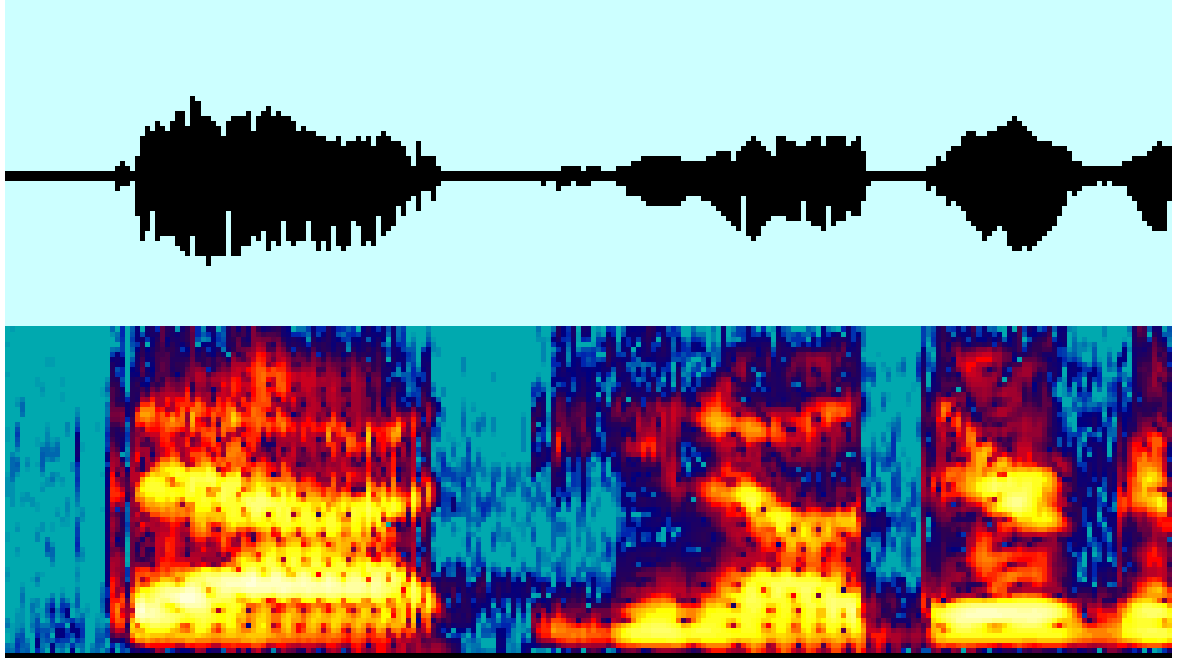


End-to-End ASR

- Since 2017
- Neural network
- Huge speech database
- Natural speech

Feature Calculation

Input:



Output:

-0.1	0.2	0.2	-6.1
0.3	0.1	0.0	-2.1
1.4	1.2	1.2	3.1
-1.2	-1.2	-1.2	2.4
2.3	4.4	4.4	1.0
2.6	2.2	2.2	2.2
…	…	…	…

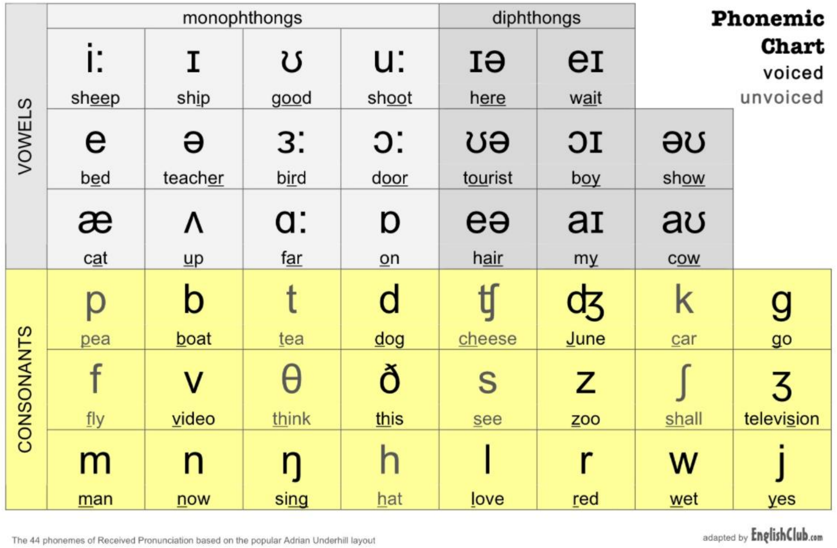
…

Typical speech features:

- 39-dim MFCC
- 80-dim filter bank output
- 400 sample points

Pronunciations

- **Phoneme:**
  - Smallest pronunciation unit to represent meanings
  - E.g. cat: K AE T K, good: G UH D G
  - Phone: The actual pronunciations of phoneme
  - There are about 44 phonemes in English



	monophthongs				diphthongs			**Phonemic Chart** voiced unvoiced
VOWELS	iː sheep	ɪ ship	ʊ good	uː shoot	ɪə here	eɪ wait		
	e bed	ə teacher	ɜː bird	ɔː door	ʊə tourist	ɔɪ boy	əʊ show	
	æ cat	ʌ up	ɑː far	ɒ on	eə hair	aɪ my	aʊ cow	
CONSONANTS	p pea	b boat	t tea	d dog	tʃ cheese	dʒ June	k car	g go
	f fly	v video	θ think	ð this	s see	z zoo	ʃ shall	ʒ television
	m man	n now	ŋ sing	h hat	l love	r red	w wet	j yes

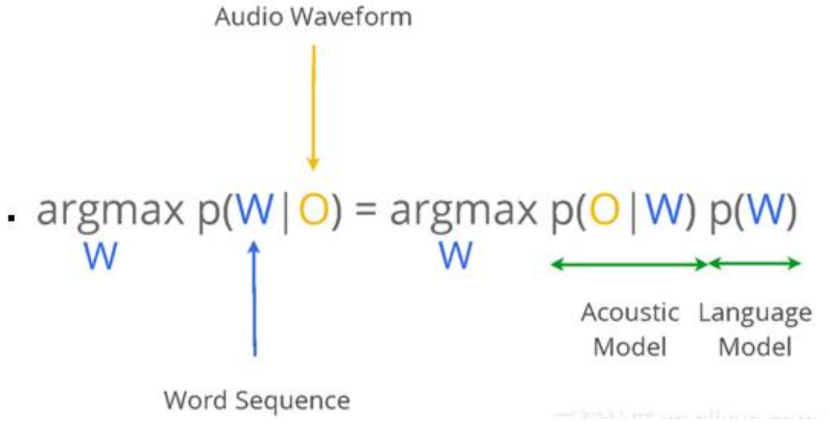
The 44 phonemes of Received Pronunciation based on the popular Adrian Underhill layout

adapted by EnglishClub.com

Source: https://www.englishclub.com/

PART 2. TRADITIONAL METHOD

Speech Recognition Framework



$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

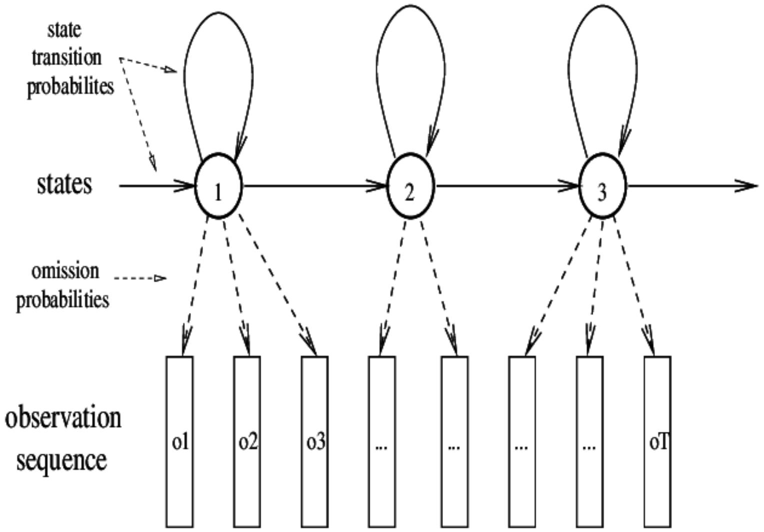
- **Acoustic model:**
  - To evaluate speech feature against pronunciations
  - How likely the speech signal sounds like the word sequence.
- **Language model:**
  - To evaluate whether the word sequence is reasonable
  - How likely the word sequence like a correct sentence

Acoustic Model

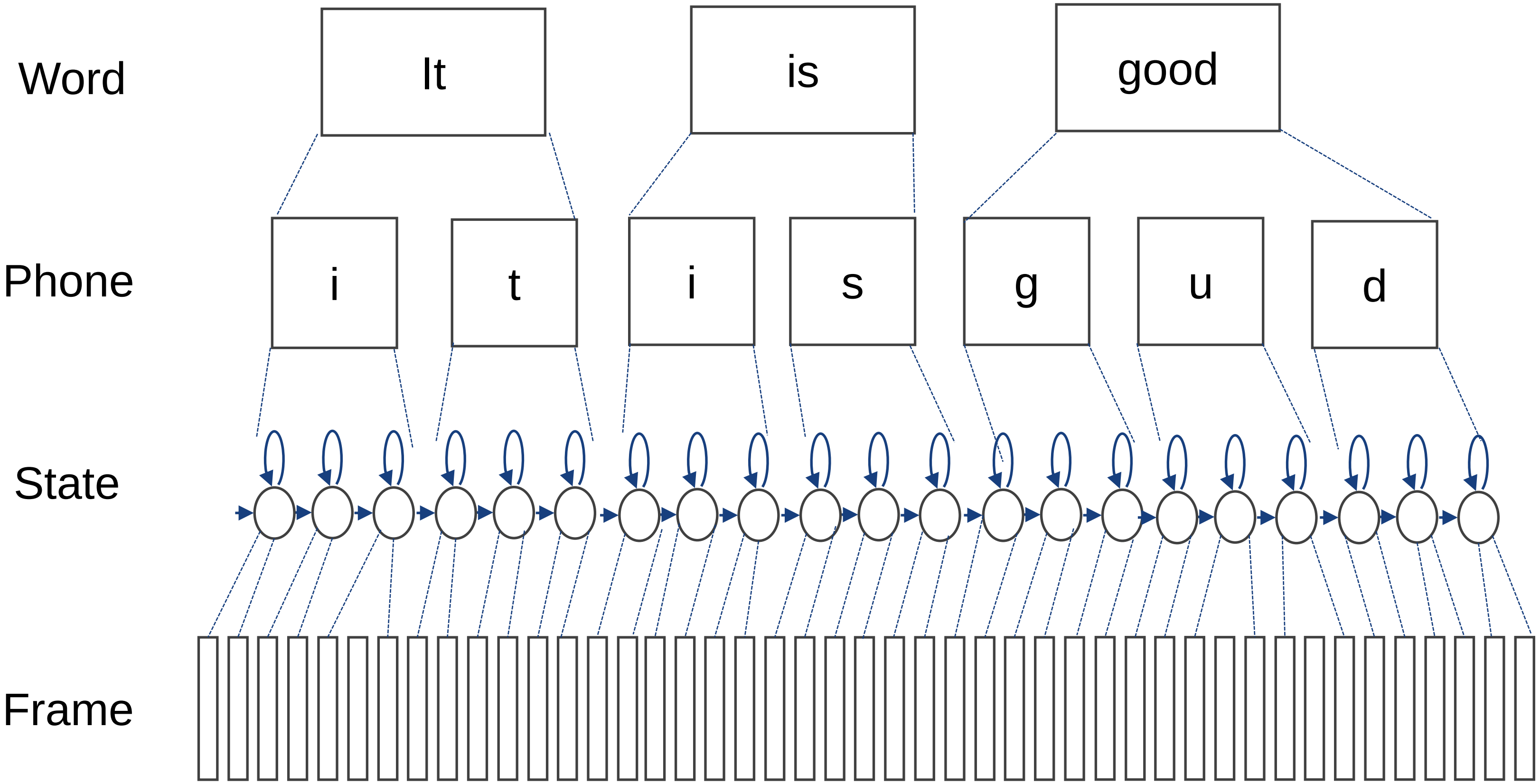
- Most common method: Hidden Markov Models

- Each phone is defined as a HMM model
- Each model contains 3-5 states.
- Speech frames are mapped to state

- Speech Recognition: To calculate **P(O|W)**

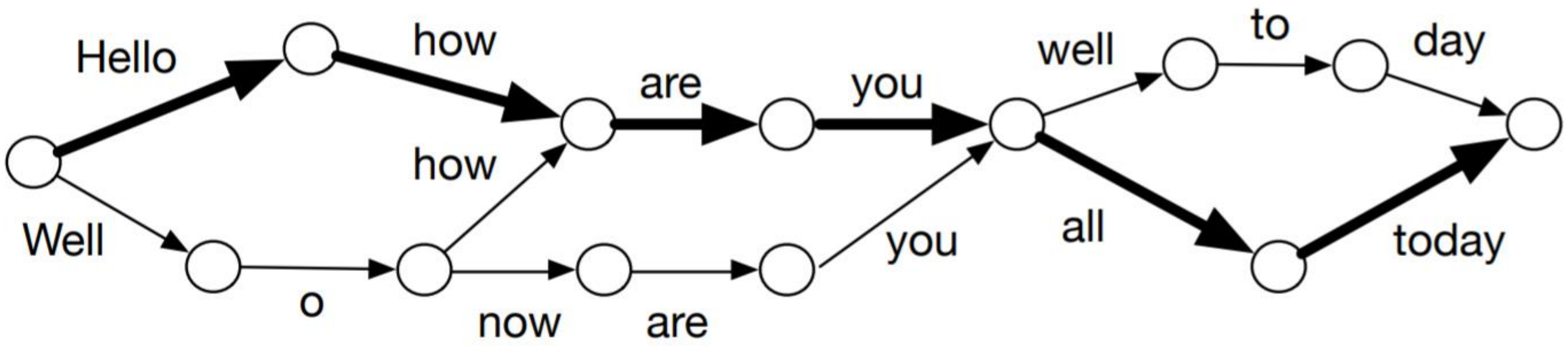


HMM for Sentence



Language Model

- To find the most like word sequence from all the possibilities.
- To calculate P(W)



NN-Grams: Unifying Neural Network and n-Gram Language Models for Speech Recognition, Babak Damavandi, Shankar Kumar, author Antoine Bruguier, INTERSPEECH 2016

Language Model

$$\text{Unigram LM} : p(w_1^N) = \prod_{n=1}^{N} p(w_n)$$

$$\text{Bigram LM} : p(w_1^N) = \prod_{n=1}^{N} p(w_n | w_{n-1})$$

$$\text{Trigram LM} : p(w_1^N) = \prod_{n=1}^{N} p(w_n | w_{n-2}, w_{n-1})$$

- **N-gram models calculate the likelihood of a word given previous word(s)**
- **Language models are trained with text corpus.**

ASR System

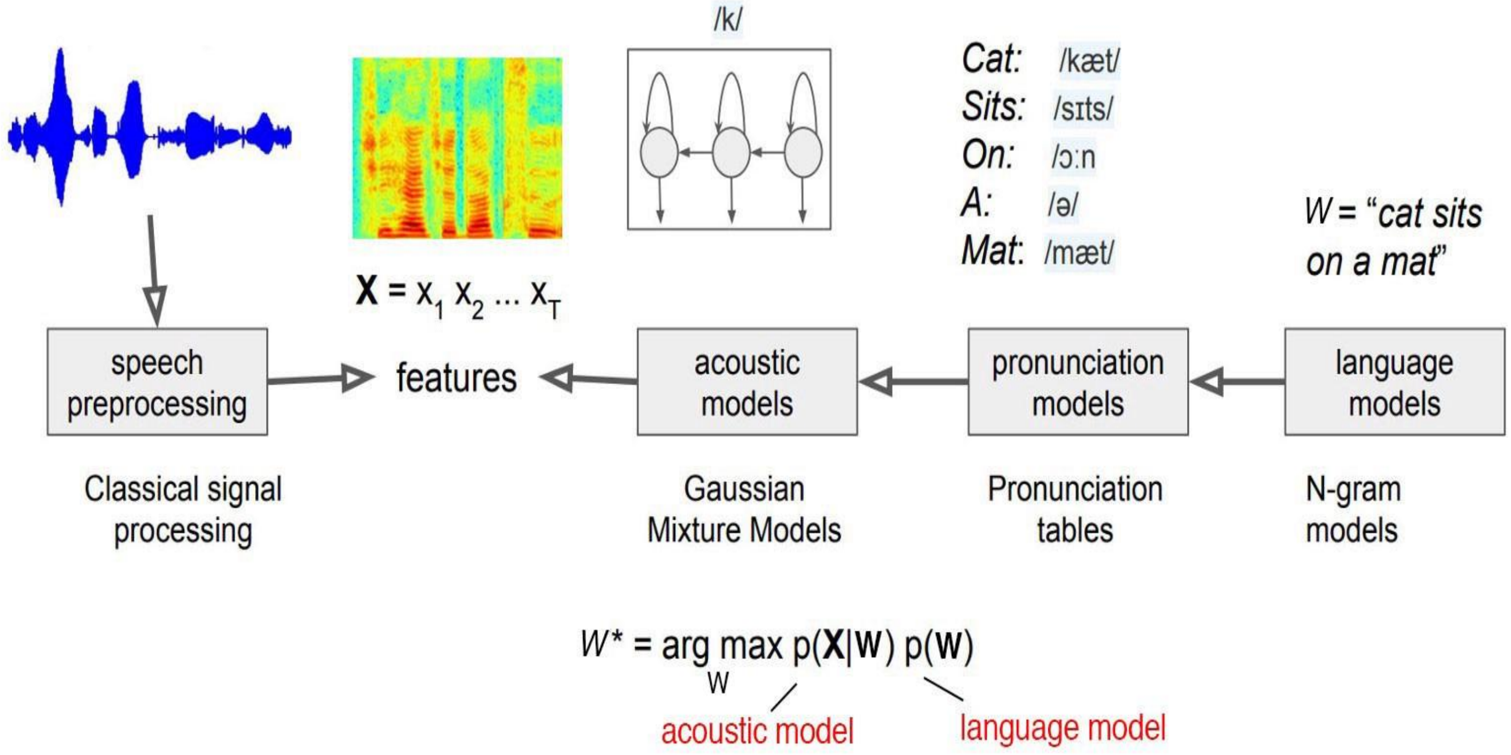


Image source: https://heartbeat.fritz.ai/the-3-deep-learning-frameworks-for-end-to-end-speech-recognition-that-power-your-devices-37b891ddc380

PART 3. RECENT ADVANCES

End-to-End Speech Recognition

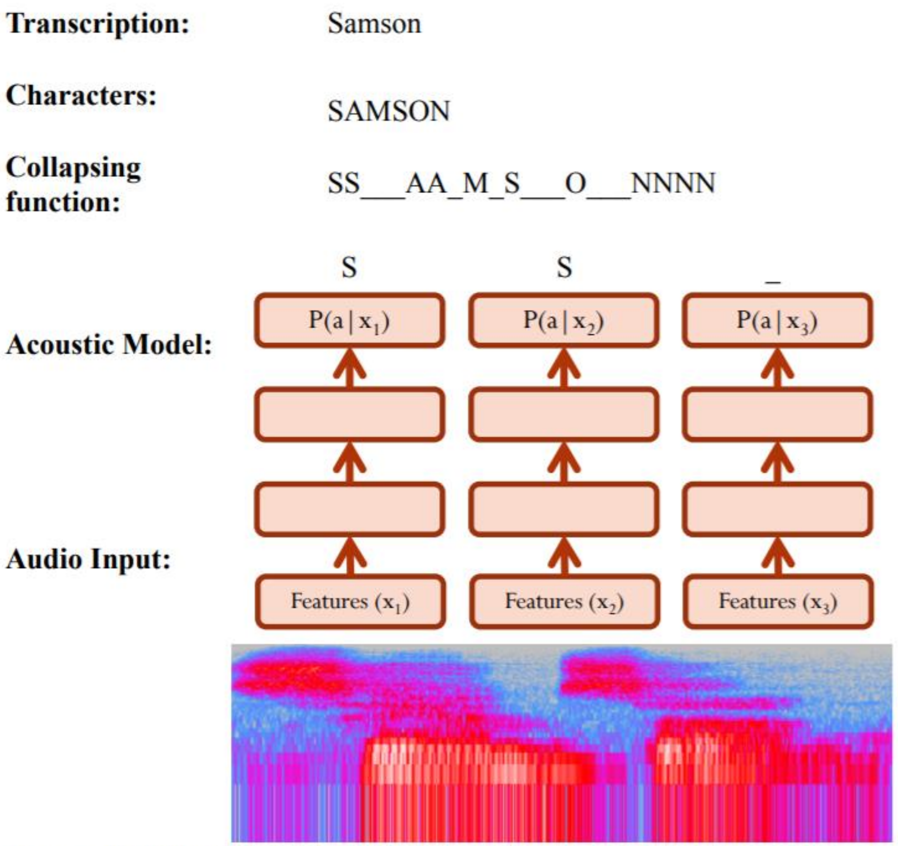
- **Connectionist Temporal Classification (CTC) method**
  - Generate a letter sequence first (with duplicated characters). Then to convert the letter sequence into text.

- **Listen, Attend, and Spell (LAS) method**
  - Encoder/decoder with attention. Directly generate sequence.

- **RNN Transducer (RNN-T)**
  - Integrate language model into prediction. Do not rely on the full sequence.

CTC-based End-to-end ASR

- Multiple modules are merged into one network for joint training.
- Sequence to Sequence model.
- It directly maps input acoustic signature sequence to the text result sequence,
- Use DNN to approximate distribution over characters.
- Simple collapsing function to generate results



Transcription: Samson

Characters: SAMSON

Collapsing function: SS___AA_M_S___O___NNNN

Acoustic Model:

Audio Input:

(Graves & Jaitly. 2014)

Example Results

YET A REHBILITATION CRU IS ONHAND IN THE BUILDING LOOGGING BRICKS PLASTER AND BLUEPRINS FOUR FORTY TWO NEW BETIN EPARTMENTS

YET A REHABILITATION CREW IS ON HAND IN THE BUILDING LUGGING BRICKS PLASTER AND BLUEPRINTS FOR FORTY TWO NEW BEDROOM APARTMENTS

THIS PARCLE GUNA COME BACK ON THIS ILAND SOM DAY SOO

THE SPARKLE GONNA COME BACK ON THIS ISLAND SOMEDAY SOON

TRADE REPRESENTIGD JUIDER WARANTS THAT THE U S WONT BACKCOFF ITS PUSH FOR TRADE BARIOR REDUCTIONS

TRADE REPRESENTATIVE YEUTTER WARNS THAT THE U S WONT BACK OFF ITS PUSH FOR TRADE BARRIER REDUCTIONS

TREASURY SECRETARY BAGER AT ROHIE WOS IN AUGGRAL PRESSED FOUR ARISE INTHE VALUE OF KOREAS CURRENCY

TREASURY SECRETARY BAKER AT ROH TAE WOOS INAUGURAL PRESSED FOR A RISE IN THE VALUE OF KOREAS CURRENCY

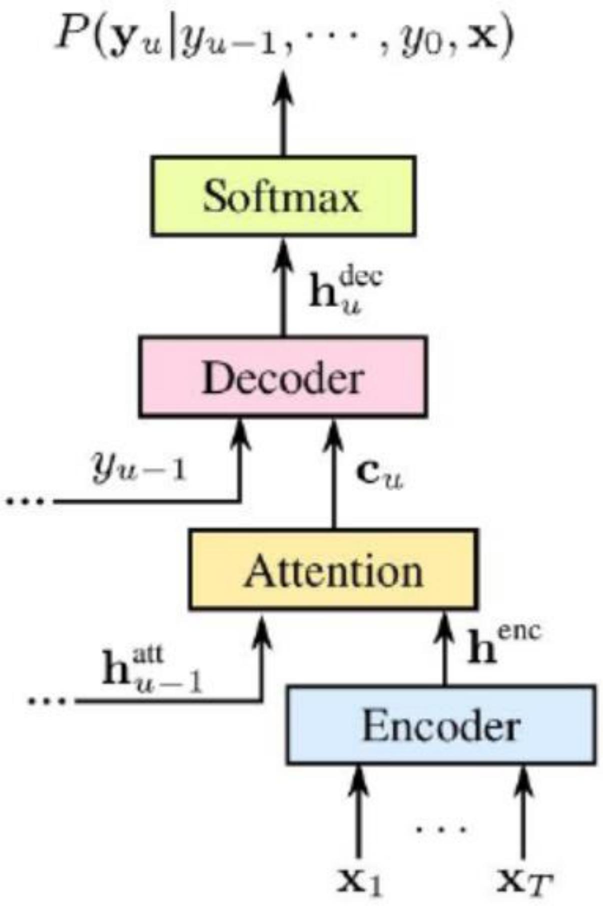
Attention-based End-to-End ASR

- **LAS method: Listen, Attend and Spell**
- **Listen: Encoder**

  Transforms input speech into higher-level representation
- **Attend: Attention**

  Identifies encoded frames that are relevant to producing current output
- **Spell: Decoder**

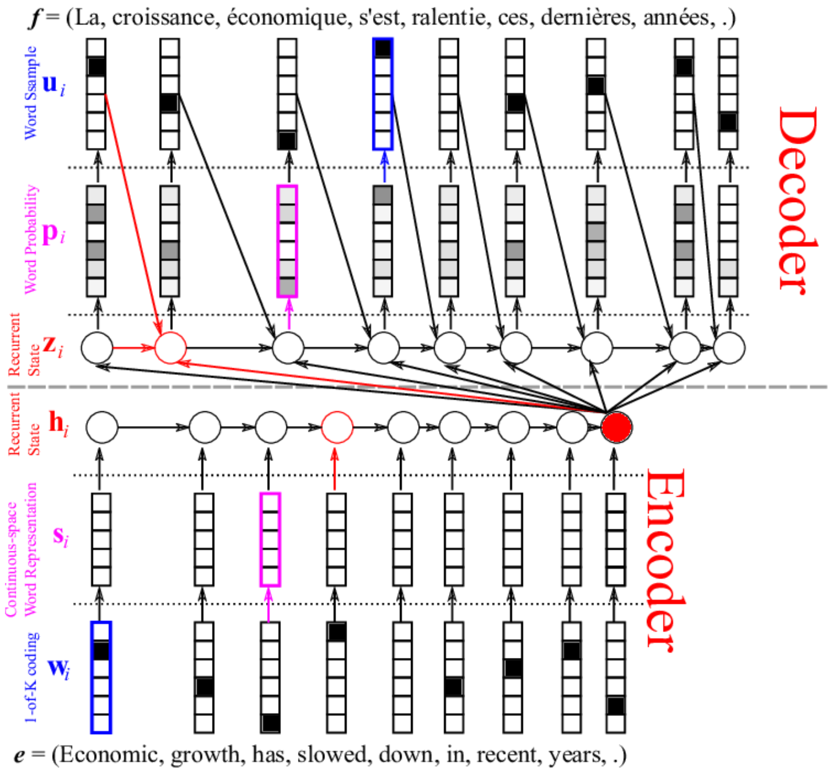
  Operates autoregressively by predicting each output token as a function of the previous predictions



$P(\mathbf{y}_u | y_{u-1}, \cdots, y_0, \mathbf{x})$
Softmax
$\mathbf{h}_u^{\text{dec}}$
Decoder
$\cdots$ y_{u-1}
$\mathbf{c}_u$
Attention
$\cdots$ $\mathbf{h}_{u-1}^{\text{att}}$
$\mathbf{h}^{\text{enc}}$
Encoder
$\mathbf{x}_1$
$\cdots$
$\mathbf{x}_T$

Attention-based End-to-End ASR

- **Recap: Sequence to Sequence Model**

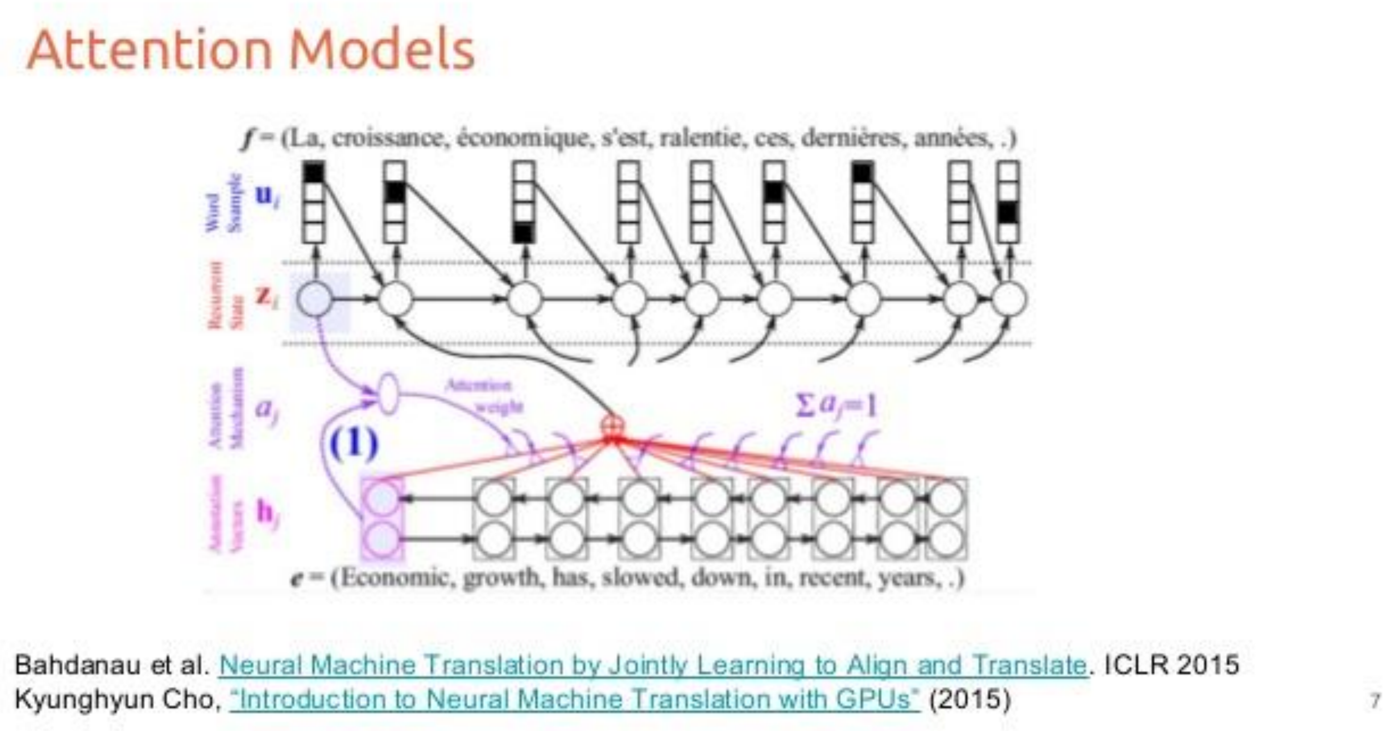

f = (La, croissance, économique, s'est, ralentie, ces, dernières, années, .)
Word Sample
$\mathbf{u}_i$
Word Probability
$\mathbf{p}_i$
Recurrent State
$\mathbf{z}_i$
Decoder
Recurrent State
$\mathbf{h}_i$
Continuous-space Word Representation
$\mathbf{s}_i$
1-of-K coding
$\mathbf{w}_i$
Encoder
e = (Economic, growth, has, slowed, down, in, recent, years, .)

Encoder: To convert the sequence into a vector

Decoder: To generate a new sequence from the encoded vector and the newly generated elements

Attention-based End-to-End ASR

- **Recap: Sequence to Sequence with Attention**



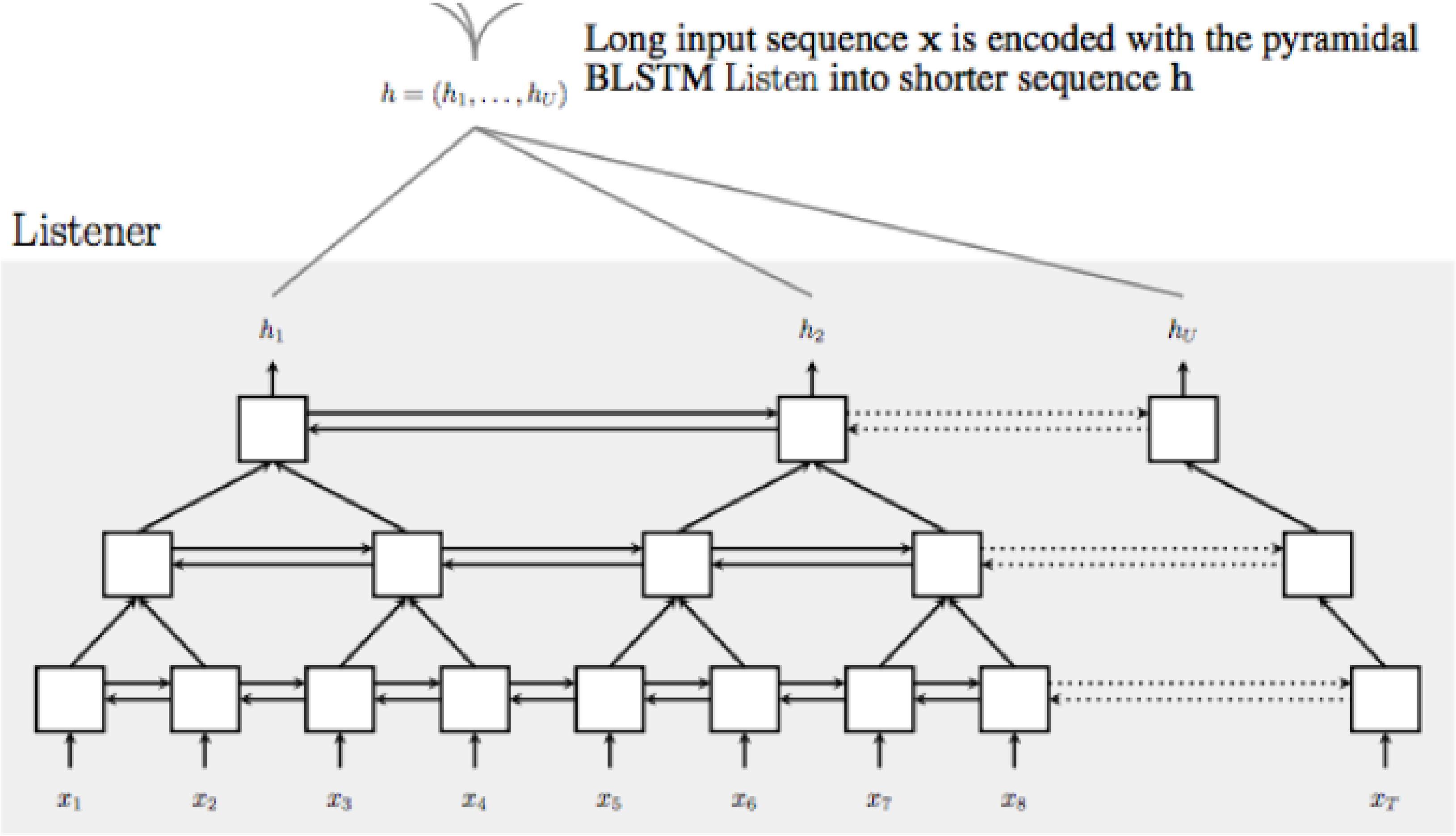
Attention Models

Bahdanau et al. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015
Kyunghyun Cho, "Introduction to Neural Machine Translation with GPUs" (2015)

7

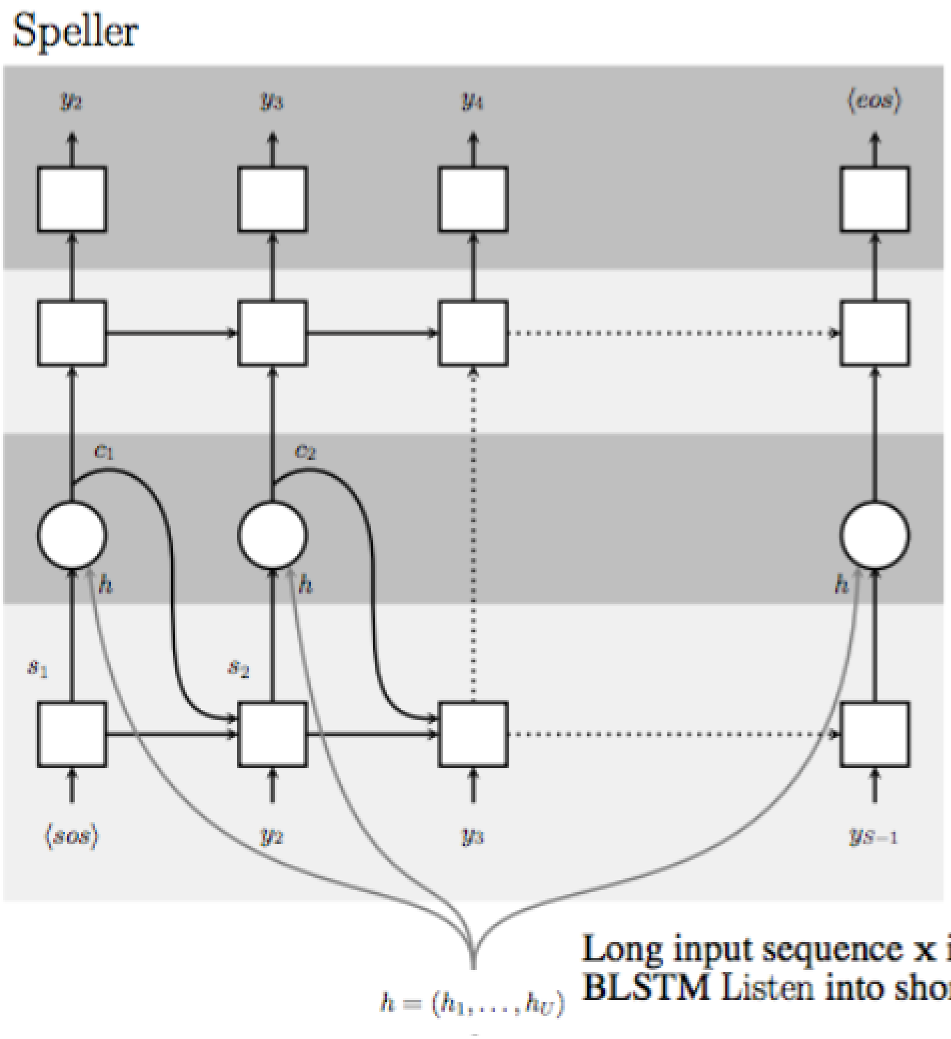
Each item generated depends on all the hidden vectors in the entire input sequence.

LAS model: Listen



Long input sequence **x** is encoded with the pyramidal BLSTM Listen into shorter sequence h

LAS model: Attend and Spell

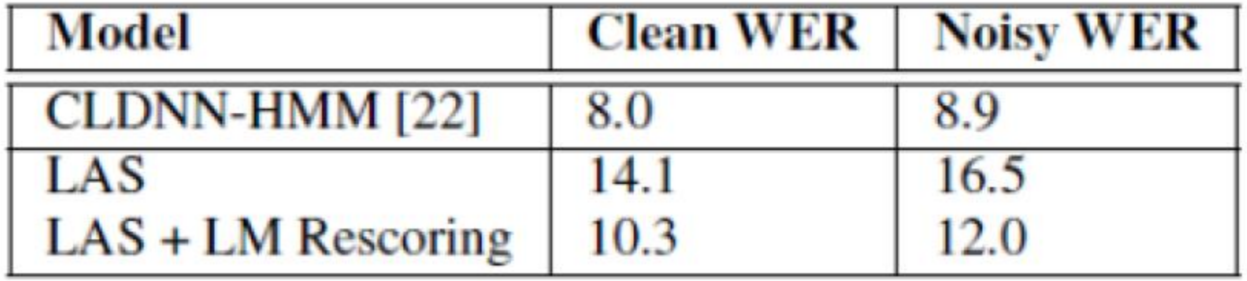


Grapheme characters y_i are modelled by the CharacterDistribution

AttentionContext creates context vector c_i from h and s_i

Long input sequence **x** is encoded with the pyramidal BLSTM Listen into shorter sequence **h**

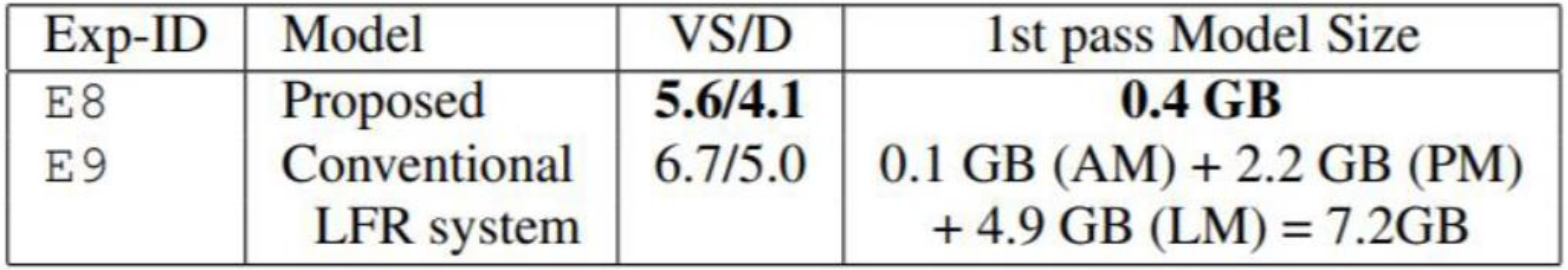
LAS model: Performance



Model	Clean WER	Noisy WER
CLDNN-HMM [22]	8.0	8.9
LAS	14.1	16.5
LAS + LM Rescoring	10.3	12.0

2000 hours

[Chan, et al., ICASSP'16]



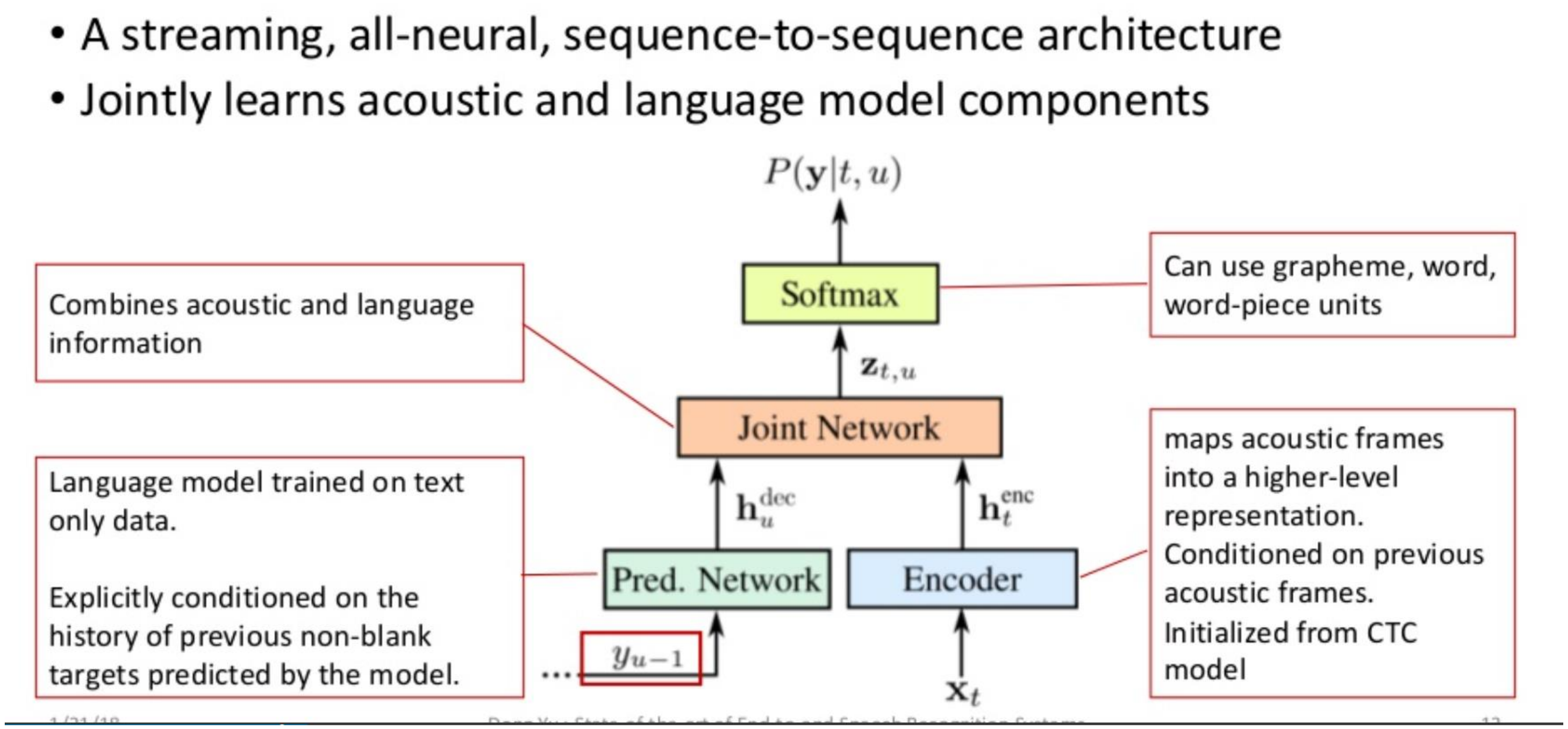
Exp-ID	Model	VS/D	1st pass Model Size
E8	Proposed	**5.6/4.1**	**0.4 GB**
E9	Conventional LFR system	6.7/5.0	0.1 GB (AM) + 2.2 GB (PM) + 4.9 GB (LM) = 7.2GB

12500 hours

[Chiu, et al., ICASSP, 2018]

It works well when trained with a large dataset.

RNN Transducer (RNN-T)



- A streaming, all-neural, sequence-to-sequence architecture
- Jointly learns acoustic and language model components

Combines acoustic and language information

Language model trained on text only data.

Explicitly conditioned on the history of previous non-blank targets predicted by the model.

Can use grapheme, word, word-piece units

maps acoustic frames into a higher-level representation. Conditioned on previous acoustic frames. Initialized from CTC model

Source: https://www.slideshare.net/BillLiu31/state-of-art-e2e-speech-recoginition-system-by-dong-yu

RNN Transducer (RNN-T)

- Properties of RNN-T

  - Do not need to process the entire input sequence to produce an output.
  - Continuously processes input samples and streams output symbols, good for speech dictation.
  - Outputs characters one-by-one, as you speak, with white spaces where appropriate.
  - Feeds predicted symbols to predict the next symbols. (Language model integrated)

PART 4. ASR SYSTEM ISSUES

How to Train an ASR system

- **Resources**
  - Speech with text transcription, big enough to cover the language, Should have word transcriptions
  - Pronunciation dictionary
  - Big text corpus
- **Models to train**
  - Acoustic models
  - Language model

Training Process

- **Dataset**
  - Modelling system needs thousand hours of speech
  - Dataset is divided into training, development and testing sets.
- **Training & Testing**
  - Train the system
  - Test on development set
  - Tune the system and repeat the process
  - At the end, test on the testing set.

Accuracy Measure

- **Usually, ASR performance is judged by the word error rate**

  ErrorRate = 100*(Subs + Ins + Dels) / Nwords

```
REF: I WANT TO   GO HOME ***
REC: * WANT TWO GO HOME NOW
SC:  D  C    S   C   C    I
100*(1S+1I+1D)/5 = 60%
```

Accuracy Measure

- **Usually, ASR performance is measured by the word error rate**
  - This assumes that all errors are equal
  - Also, a bit of a mismatch between optimization criterion and error measurement

- **Task specific measures sometimes used**
  - Task completion
  - Concept error rate

Data Collection

- **Normally hundreds or thousands of people’s voice data are needed**
- **Gender balanced; pronunciation balanced; device/channel balanced;**
- **Expensive to collect and transcribe speech data.**
- **Morden systems are normally trained with thousands of hours speech data. (Some recent ones using 60000 hours or more)**
- **Data augmentation methods are often used to increase the size of training database.**

Speaker Dependent System

- **Speaker dependent/independent**
  - Current ASR systems are normally developed for any user.
  - But ASR system can be adapted to a particular speaker.
- **Speaker dependent system:**
  - System is built for a particular speaker only.
  - Relatively easy to achieve higher accuracy.
- **Speaker independent system:**
  - System works for any new speaker.
  - Difficult to achieve high accuracy

Considerations in ASR Deployment

- **Free text or limited text**
  - Free text: large vocabulary free text recognition

    Examples: dialog system, dictation, etc
  - Voice commands: Limited number of commands. Examples: ASR for embedded system.
- **Cloud or embedded**
  - Cloud: Speech recognition is on cloud. Normally the recognition model is huge. Most of free text recognition engines are on cloud.
  - Embedded system: Small system data size.

ASR Deployment: Signal consideration

- **Distance**
  - Near field: Smartphone, PC, etc
  - Far field: Microphone is far away from speaker.

    Example: Echo, etc
- **Channel/sampling rate**
  - Digital system: Digital system like smartphone, sampling rate is 16KHz.
  - Telephony: The sampling rate is 8KHz for tranditional telephony systems.

ASR deployment：System consideration

- **Multi-thread processing**
  - Real-time factor: Time needed to process 1 second speech. Real-time factor 0.1 means 1 CPU can support 10 users.
  - Memory: System data can be shared among running instances. E.g. model, lexicon, etc. Working data need additional memory for each user. (E.g. intermediate data generated for the user in recognition process)
- **Response time**
  - Time used for data transfer: data sent to server, result sent to client.
  - Delay in speech recognition engine for real time processing.

More Challenging Scenarios

- **Out-of-Vocabulary words**
  - New word or names not in system lexicon
- **Spontaneous speech**
  - Informal natural speech, pauses, correction, repeating, etc.
- **Cross-talking speech**
  - Two or more people talking at the same time.
- **Code-switch speech recognition**
  - Mix of different languages in speech

Speech Recognition Toolkit

- **Kaldi**
  - Kaldi is a toolkit for speech recognition,
  - For use by speech recognition researchers and professionals.
  - Support many of the advanced features.
- **HTK**
  - A proprietary software toolkit for handling HMMs
- **EPSnet**
  - End-to-end speech processing toolkit

Integrated Speech Recognition

- Windows system
- MacOS
- iPhone
- Android system

Cloud Services

- Google Speech Recognition
- Google Cloud Speech API
- Wit.ai
- Microsoft Bing Voice Recognition
- Houndify API
- IBM Speech to Text

Examples of Speech Application

- Telephony system
- Medical records
- Court hearing transcription
- Speech analytics for call center
- Voice assistant/Smart speaker
- Voice control devices
- In-car application
- Air flight traffic control

PART 5. ASR PROGRAMMING

Hands-on Session

- **Open source python speech recognition**
  - Install: pip install SpeechRecognition
  - Library for performing speech recognition
  - Supported engines:
    - CMU Sphinx (works offline)
    - Google Speech Recognition
    - Google Cloud Speech API
    - Wit.ai
    - Microsoft Bing Voice Recognition
    - Houndify API
    - IBM Speech to Text
    - Snowboy Hotword Detection (works offline)

Hands-on Session

- **Examples: SpeechRecognition**

  https://pypi.org/project/SpeechRecognition/

  - Recognize speech input from the microphone
  - Transcribe an audio file
  - Save audio data to an audio file
  - Calibrate the recognizer energy threshold for ambient noise levels
  - Listening to a microphone in the background

Hands-on Session

- **Google cloud speech recognition**
  - https://cloud.google.com/speech-to-text
  - A chargeable service from google cloud.
  - Steps：
    - Set up a Cloud Console project. (create project, enable API, create service account, download private key)
    - Set environment variable GOOGLE_APPLICATION_CREDENTIALS
    - Install and initialize the Cloud SDK.

  https://cloud.google.com/speech-to-text/docs/quickstart-protocol?hl=en_GB

Thank you!

mhdong@i2r.a-star.edu.sg