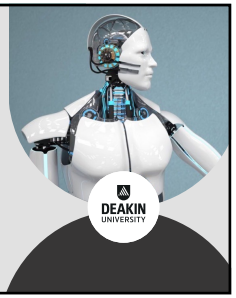
SIT330-770: Natural Language Processing

Week 9 – Speech Processing & ASR

Dr. Mohamed Reda Bouadjenek

School of Information Technology, Faculty of Sci Eng & Built Env

reda.bouadjenek@deakin.edu.au



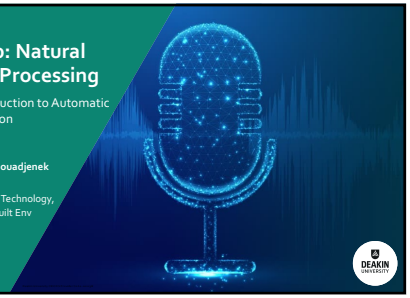
1

SIT330-770: Natural Language Processing

Week 9. 1 - Introduction to Automatic Speech Recognition

Dr. Mohamed Reda Bouadjenek

School of Information Technology, Faculty of Sci Eng & Built Env



2

What is speech recognition?

- Speech-to-text transcription
  - Transform recorded audio into a sequence of words
  - Just the words, no meaning…. But do need to deal with acoustic ambiguity: "Recognise speech?" or "Wreck a nice beach?"
  - Speaker diarization: Who spoke when?
  - Speech recognition: what did they say?
  - Paralinguistic aspects: how did they say it? (timing, intonation, voice quality)
  - Speech understanding: what does it mean?

3

Applications of ASR

- Dictation
- Language learning
- Smart speakers (Alexa, Siri)
- Accessibility for hearing impaired
- Voice command
- Automatic captioning
- Audio indexing
- Machine translation
- Meeting understanding and summarization
- Call center analysis
- TV remote
- …

Bixby · Hey Siri · 天猫精灵 · alexa · 小爱同学 · Cortana · DUEROS 对话式人工智能操作系统 · Google Assistant

4

Enable ChatGPT with voice input/output

5

Why is speech recognition difficult?

- Several sources of variation
  - **Size**
    - Number of word types in vocabulary, perplexity
  - **Speaker**
    - Tuned for a particular speaker, or speaker-independent? Adaptation to speaker characteristics
  - **Acoustic environment**
    - Noise, competing speakers, channel conditions (microphone, phone line, room acoustics)
  - **Style**
    - Continuously spoken or isolated? Planned monologue or spontaneous conversation?
  - **Accent/dialect**
    - Recognise the speech of all speakers who speak a particular language
  - **Language spoken**
    - There are many languages beyond English, Mandarin Chinese, Spanish, . . . What is the difference between a dialect and a language?

6

SIT330-770: Natural Language Processing

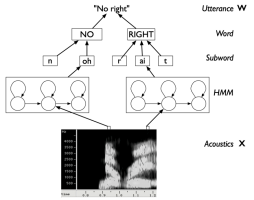
Week 9. 2 - Statistical modeling for Automatic Speech Recognition

Dr. Mohamed Reda Bouadjenek

School of Information Technology, Faculty of Sci Eng & Built Env

7

Hierarchical modelling of speech



8

Statistical 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

$$W^* = \underset{W}{\operatorname{argmax}}\, P(W|X)$$

- Applying Bayes' Theorem:

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

$$\propto P(X|W)P(W)$$

$$W^* = \underset{W}{\operatorname{argmax}}\, P(X|W)\, P(W)$$

Language model

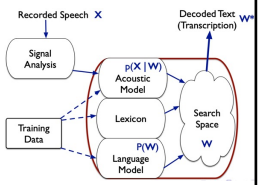
Acoustic model

9

Speech Recognition Components

$$W^* = \underset{W}{\operatorname{argmax}}\, P(W|X)$$

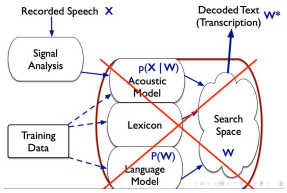
- Use an acoustic model, language model, and lexicon to obtain the most probable word sequence **W*** given the observed acoustics **X**



10

Alternative approach: End-to-end systems

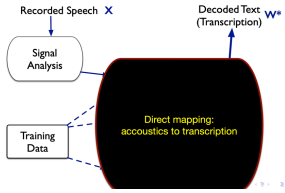
- Directly model transforming an input acoustic sequence into an output word or character sequence



11

Alternative approach: End-to-end systems

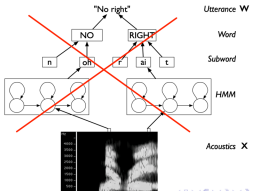
- Directly model transforming an input acoustic sequence into an output word or character sequence



12

Alternative approach: End-to-end systems

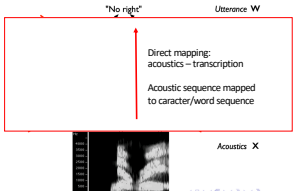
- Directly model transforming an input acoustic sequence into an output word or character sequence



13

Alternative approach: End-to-end systems

- Directly model transforming an input acoustic sequence into an output word or character sequence



14

SIT330-770: Natural Language Processing

Week 9. 3 - Evaluation Metrics

Dr. Mohamed Reda Bouadjenek

School of Information Technology,
Faculty of Sci Eng & Built Env

15

Evaluation Metrics

- Reference:
  - The quick brown fox jumped over the lazy dog
- Hypothesis:
  - The quick brown fox jumps over ---- lazy dog too
- Word error rate:
  - $WER = D+S+I\ N$
    - D: number of deleted words
    - S: number of subsituted words
    - I: number of inserted words
    - N: number of reference words
- Readability: whether the recognized text is easy to read by human.

16

SIT330-770: Natural Language Processing

Week 9. 4 - Deep learning for ASR

Dr. Mohamed Reda Bouadjenek

School of Information Technology,
Faculty of Sci Eng & Built Env

17

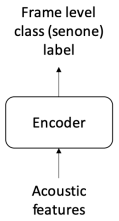
Deep learning for ASR

- Hybrid system: only replace HMM/GMM acoustic model with neural networks
- End-to-end ASR: replace the whole ASR system with neural works

18

Hybrid acoustic model

- Replace the generative HMM/GMM with a discriminative neural networks
- HMM/GMM models $p(o_t|s_t)$
- Hybrid models $p(s_t|o_t)$
- Common practices
  - Train an HMM/GMM first
  - Use it to align the label (senone sequences) to the feature sequence.
  - Train neural networks to predict frame level senone labels



19

Encoder Structures

- DNN
- CNN
- LSTM
- Transformer
- Or any combination of them

20

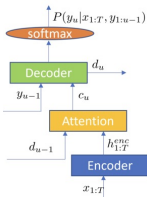
End-to-end ASR

- End-to-end ASR systems try to do ASR with a single model
- Three main approaches
  - Connectionist Temporal Classification
  - RNN Transducers
  - Sequence-to-Sequence

21

Sequence-to-sequence (S2S)

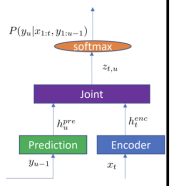
- S2S is also called attention encoder decoder (AED)
- Encoder: similar to acoustic model
- Attention: alignment model
- Decoder: similar to pronunciation and language model
- Offline model



22

RNN Transducers (RNN-T)

- Called RNN-T because originally RNN is used as the encoder model structure.
- Newer models uses transformers or conformers as encoder
- A native streaming model



23

SIT330-770: Natural Language Processing

Week 9. 5 - The alignment problem

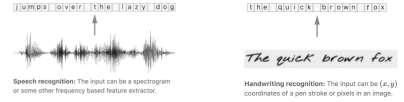
Dr. Mohamed Reda Bouadjenek

School of Information Technology, Faculty of Sci Eng & Built Env

24

The alignment problem

- We have a data set of speech, handwriting, other sequential data and the corresponding transcripts
- **Problem: we don't know how the outputs align to the inputs**
  - **i.e., which frame(s) of the input correspond to which output frame**



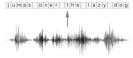
Speech recognition: The input can be a spectrogram or some other frequency based feature extractor.

Handwriting recognition: The input can be (x, y) coordinates of a pen stroke or pixels in an image.

25

The alignment problem: Naïve solutions

- We could devise a rule like "one character corresponds to ten inputs".
  - But people's rates of speech vary, so this type of rule can always be broken.
- Another alternative is to hand-align each character to its location in the audio.
  - May work well, but we'd know the ground truth for each input time-step
  - For any reasonably sized dataset this is prohibitively time consuming.



Speech recognition: The input can be a spectrogram or some other frequency based feature extractor.

26

Solution: Connectionist Temporal Classification (CTC) is a way to get around not knowing the alignment between the input and the output

27

Problem definition

- Given:
  - A sequence $X=[x_1,x_2,\ldots,x_T]$ (audio)
  - The corresponding output sequence $Y=[y_1,y_2,\ldots,y_U]$ (transcript)
- We want to find an accurate mapping from X to Y
- Challenges:
  - Both X and Y can vary in length
  - The ratio of the lengths of X and Y can vary.
  - We don't have an accurate alignment (correspondence of the elements) of X and Y
- The CTC algorithm overcomes these challenges and for a given X it gives an output distribution over all possible Y
  - We can use this distribution either to *infer* a likely output or to assess the *probability* of a given output.

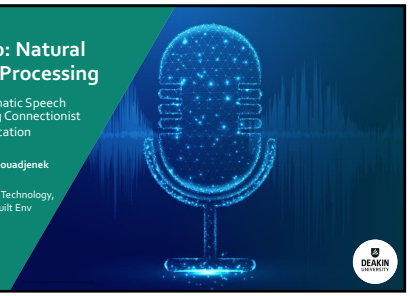
28

SIT330-770: Natural Language Processing

Week 9. 6 -Automatic Speech Recognition using Connectionist Temporal Classification

Dr. Mohamed Reda Bouadjenek

School of Information Technology, Faculty of Sci Eng & Built Env



29

The algorithm: Alignment

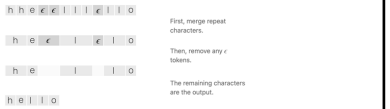
- Assume the input has length six and Y = [c, a, t]. One way to align X and Y is to assign an output character to each input step and collapse repeats
- This approach has two problems:
  - It doesn't make sense to force every input step to align to some output
  - We have no way to produce outputs with multiple characters in a row.
    - The alignment **[h, h, e, l, l, o]** collapses to **"helo"**

$x_1\ x_2\ x_3\ x_4\ x_5\ x_6$	input (X)
c c a a a t	alignment
c a t	output (Y)

30

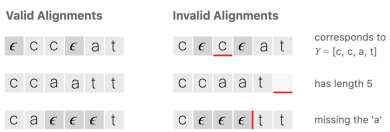
The algorithm: CTC Alignment

- CTC introduces a new token ϵ called the blank token
- The ϵ token doesn't correspond to anything
- We allow any alignment which maps to Y after merging repeats and removing ϵ tokens:



31

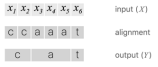
The algorithm: CTC Alignment Examples



32

The algorithm: CTC Alignment Properties

- The allowed alignments between X and Y are monotonic.
  - If we advance to the next input, we can keep the corresponding output the same or advance to the next one.
- The alignment of X to Y is many-to-one.
  - One or more input elements can align to a single output element but not vice-versa.
- The length of Y cannot be greater than the length of X.



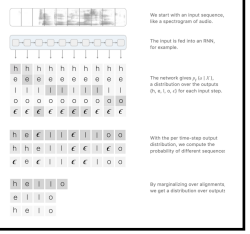
33

The algorithm: Loss Function

- The CTC alignments gives us a probability of an output sequence
- The CTC objective for a single (X,Y) pair is:

$$p(Y \mid X) = \sum_{A \in \mathcal{A}_{X,Y}} \prod_{t=1}^{T} p_t(a_t \mid X)$$

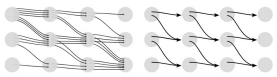
The CTC conditional **probability** — **marginalizes** over the set of valid alignments — computing the **probability** for a single alignment step-by-step.



34

The algorithm: Loss Function

- The CTC loss can be very expensive to compute.
  - A brute force approach that computes the score for each alignment is expensive
    - There can be a massive number of alignments.
- We can compute the loss faster with a **dynamic programming** algorithm
  - If two alignments have reached the same output at the same step, they can be merged



35

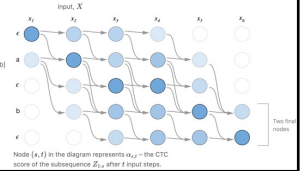
The algorithm: Loss Function

- Example of the computation performed by the dynamic programming algorithm
- Every valid alignment has a path in this graph.
- For a training set D, the loss function is:

$$\sum_{(X,Y) \in \mathcal{D}} -\log\, p(Y \mid X)$$

output $Y = [a, b]$

- The CTC loss function is differentiable since it's just sums and products of probabilities



36

The algorithm: Inference

- Find a likely output for a given input by solving:

$$Y^* = \underset{Y}{\operatorname{argmax}}\ p(Y \mid X)$$

- Need to settle for an approximate solution, too expensive to search for the true max
- One heuristic is to take the most likely character at each output

37

37

The algorithm: Inference

$$Y^* = \underset{Y}{\operatorname{argmax}}\ p(Y \mid X)$$

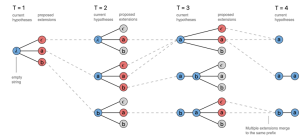
- Problems?
  - Does not take into account that the same output Y could be produced by two different alignments
  - [a,a] and [a,a,a] individually have lower probability than [b,b], but combined higher and they collapse to [a]
  - With this heuristic, [b] gets picked

38

38

The algorithm: Inference

- A better heuristic is to use modified beam search $Y^* = \underset{Y}{\operatorname{argmax}}\ p(Y \mid X)$
- Can exchange speed for asymptotically better solution



The CTC beam search algorithm with an output alphabet {c, a, t} and a beam size of three.

39

39