Transformers in Language and Speech Processing

Part II – Transformers in Automatic Speech Recognition

Marc Evrard, Camille Guinaudeau, François Yvon

LISN — CNRS and Université Paris-Saclay

2020-2021

Outline

Introduction to ASR

1. Spoken communication
2. Historical perspective
3. Statistical and neural-based
4. End-to-end approach

Transformers for ASR

1. Attention for speech
2. Self-attention for speech
3. Transformer-based ASR models
4. Self-supervised pre-training for speech

Oral communication

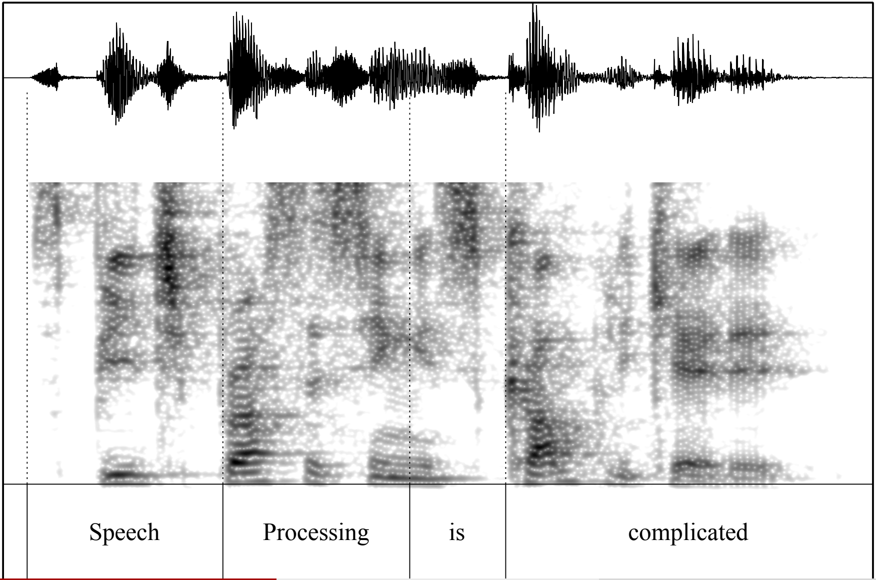
Interest of spoken communication for human-machine interaction

- Means of communication between humans
    - More natural
    - We're all experts
    - Fast: 150 wpm vs 20-50 wpm on keyboards
    - Specific needs:
        - telephony
        - help for the disabled
    - Additional modality
- Applications of automatic speech processing
    - Encoding (vocoder: telecommunications)
    - Text-to-speech synthesis
    - Speech recognition

What to recognize in speech?

- A lot of information is present in a speech signal:
  - **Speaker recognition**: Who spoke?
  - **Transcription**: What was said?
  - **Language identification**: Which language?
  - **Recognition of emotions**: In what psychological state?
- Non-verbal aspect of the voice:
  - Timbre, vocal quality, disfluencies (filler, stutter, etc.)
  - Prosody: melody + intensity + rhythm + . . .

Complexity of speech I



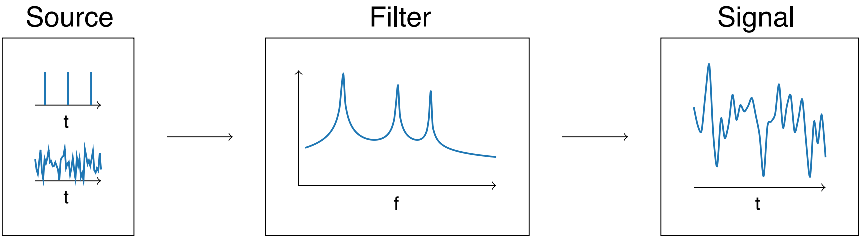
Complexity of speech II

Signal resulting from production, perception, and understanding constraints

- Signal continuity, coarticulation:
  - **no obvious segmentation**
- **Temporal distortions**:
  - variable rate
- **Context variability**:
  - inter- and intra-speakers, acoustic conditions
- Homophonies:
  - **different** transcriptions, **identical** pronunciation

60's: Rule-based approach (Dawn of AI)

- **Gunnar Fant**: **source-filter model** of speech production
- IBM: 16-word *Shoebox* machine's speech recognition
- Linear predictive coding (LPC), a speech coding method (Nagoya University and NTT)†

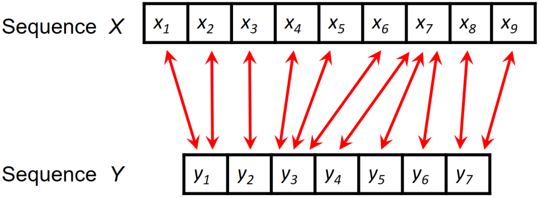


e(n) *h(n)* *x(n)*

†Fig. from https://ccrma.stanford.edu/~hskim08/lpc

70’s: Pattern recognition (Isolated words)

- DARPA funded: Carnegie Mellon’s *Harpy* speech-understanding system (understand 1011 words)
- DTW: recognition of isolated words, success of the *engineer* approach†



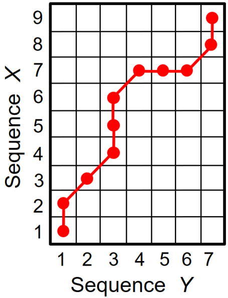


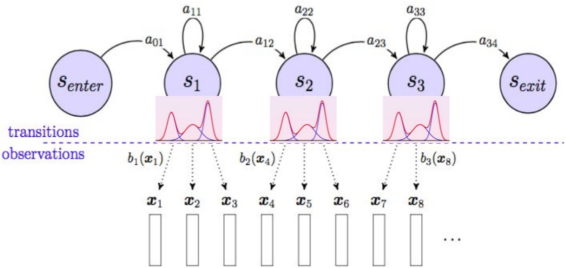
Figure 3.12 from [Müller, FMP, Springer 2015]

†Fig. from https://www.audiolabs-erlangen.de

80's: Statistical approaches (Continuous speech)

- **HMMs** based recognition:[†] James and Janet Baker (Dragon systems)
- **Fred Jelinek** (IBM): *Tangora*
  (HMM-based voice-activated typewriter, 20,000-word)

Anytime a linguist leaves the group, the recognition rate goes up



[†] Laurent Besacier, ASR-intro 2019, Université Grenoble Alpes

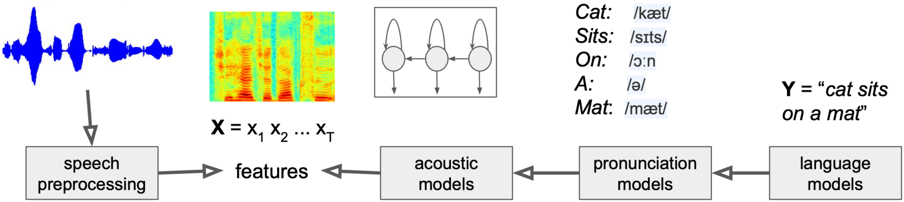
90's: International evaluation campaigns

- DARPA/NIST international assessment campaigns
- Dragon Dictate, a consumer product released in 1990
  - Lawrence Rabiner (AT&T): Voice Recognition Call Processing (VRCP) service to route telephone calls without human operators
- Introduction of the n-gram language model
- Development of neural architectures
  (that will allow for speech representation):
  - CNN: Convolutional neural networks (LeCun et al. 1995)
  - LSTM: Long short-term memory (Hochreiter et al. 1997)
  - Gradient descent for neural networks (LeCun et al. 1998)

Since 2000: The rise of DNNs

- **2000's: Larger corpora, rise of DNN**
  - DARPA: Funded the collection of the Switchboard telephone speech corpus
- **2010's: Introduction of DNN**
  - (Deep) neural networks (Hinton et al. 2012)
  - Speaker independence
    (systems used to require adaptation training for new speakers)
  - Distribution of consumer applications
    (e.g., Google, Apple, Nuance)
- **2017:**
  - **Human parity milestone** of transcribing conversational telephony speech (Microsoft)
    - CNN-BLSTM acoustic model
    - Character-based LSTM language models

Statistical-based ASR



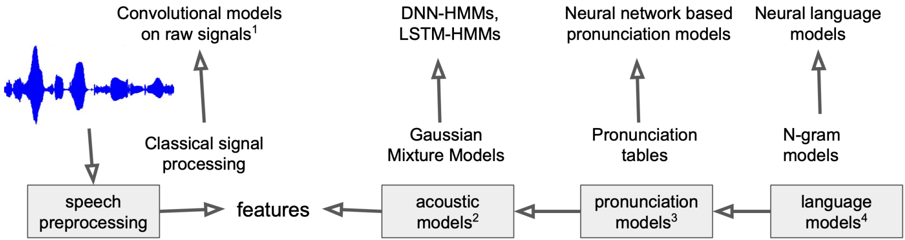
- The various modules are specialized and rely on techniques specific to their domain†
- The acoustic, pronunciation, and language models specify explicitly:

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

Aim Find the most likely text sequence Y^* that produced the given audio features X

†Fig. from Stanford cs224n Lecture 12 (2017)

Neural-based ASR†



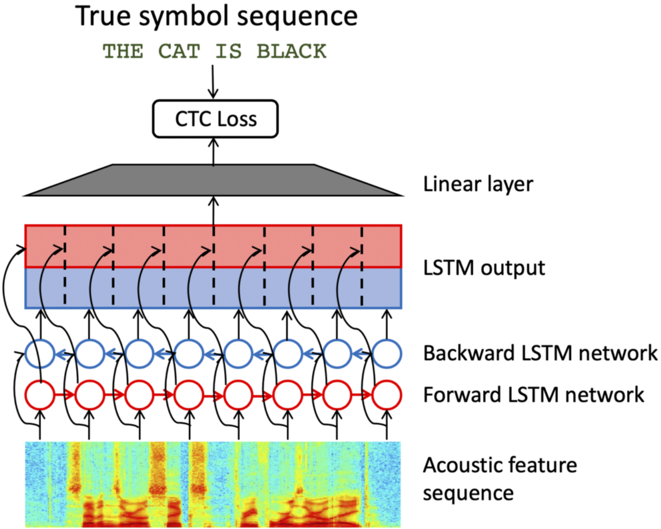
1. Jaitly et al. (2011) *Learning a better representation of speech sound waves using RBMs*
2. Hinton et al. (2012) *DNN for acoustic modeling in speech recognition*
3. Rao et al. (2015) *Grapheme-to-phoneme conversion using LSTM*
4. Mikolov et al. (2010) *Recurrent neural network-based language model*

- Each component is trained **independently** (different objective functions)
- Errors within each component may **amplify errors** in the others

Solution: Train a global **end-to-end** model (Graves et al. 2014)

†Fig. from Stanford cs224n Lecture 12 (2017)

LSTM-based†



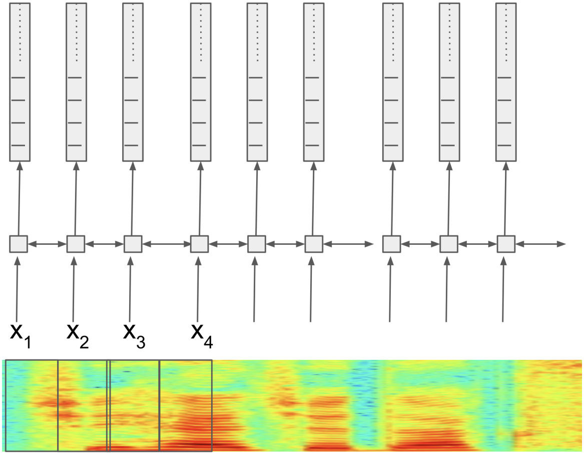
†Fig. from Audhkhasi et al. (2019)

Connectionist Temporal Classification Loss

- **CTC: Loss function** associated with RNNs (Graves et al. 2006)
- Tailored for **sequence modeling** where **timing differs** between the **input** and **output** sequences
  - E.g., typically used for modeling phonemes
- Find the **best path** through a **matrix of softmax** outputs at each frame (targeting the whole dictionary and a blank token)
- Solved efficiently through a **dynamic programming** algorithm
- **Gradients** can be calculated from the CTC scores and be back-propagated to update the neural network weights
- CTC is **independent** of the underlying neural network structure

CTC Loss I

- Compute the softmax through the network for each feature frame[†]

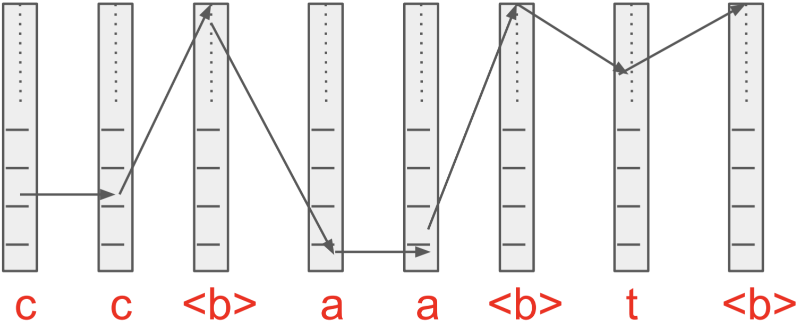


[†]Fig. from Stanford cs224n Lecture 12 (2017)

CTC Loss II

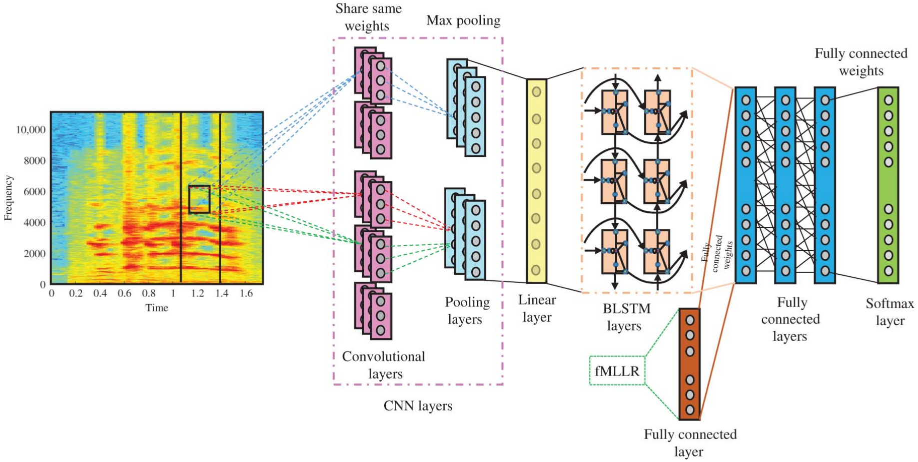
- Find the best path through the softmax at each frame (for "cat")[†]

$$Y^* = \operatorname*{argmax}_{Y} P(Y \mid X)$$



[†]Fig. from Stanford cs224n Lecture 12 (2017)

CNN-LSTM-hybrid based†



†Fig. from Passricha et al. (2020)

Attention in NMT

- Core idea: On each step of the decoder, use a **direct connection encoder** to **focus on a particular part** of the source sequence
- Main aims of **attention**:†
  - Provide a solution to the seq-to-seq **bottleneck** problem
    - Raymond Mooney (2014): *You can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector!*
    - Decoder can **look directly** at the source, bypassing the bottleneck
  - Help with the **vanishing gradient** problem
    - Provides shortcuts to distant states
  - Provides some **interpretability**
    - Can inspect what the decoder was focusing on
    - We learn a structure (soft alignment), without an explicit loss

† Inspired by Stanford cs224n Lecture 7 (2021)

Attention in general

- General definition of attention:
  - Technique to compute a **weighted sum of vector values**, dependent on a **vector query**
- *The query attends to the values*†
  - E.g., in the seq2seq + attention model:

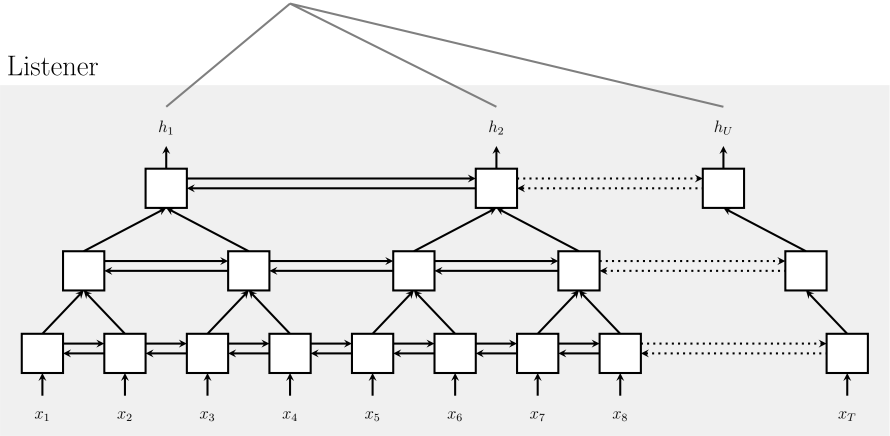
    Query (**decoder** hidden state) $\rightarrow$ **Values** (**encoder** hidden states)
  - Intuition: **Attention** is
    - **Weighted sum**: **Selective summary** of the information contained in the values (the query determines which values to focus on)
    - Way to obtain a **fixed-size representation**: of a set of representations (**values**) dependending on some other representation (the **query**)

† Inspired by Stanford cs224n Lecture 7 (2021)

Attention in Speech: Listen Attend and Spell

- Listen Attend and Spell (Chan et al. 2016)
    - NN that learns to transcribe speech utterances to characters
- Learns all components of a speech recognizer jointly
  (Unlike traditional DNN-HMM models)
    - **Listener**: Pyramidal RNN encoder (inputs: filter bank **spectra**)
    - **Speller**: Attention-based RNN decoder (outputs: **characters**)
- Attention method:
    - Speller LSTM produces a probability distribution (softmax) over the next character conditioned on all previous characters (for every output step)
- Results on a Google voice search task subset:
    - WER = 14.1% (without dictionary or LM)
    - WER = 10.3% (with LM rescoring over the top 32 beams)

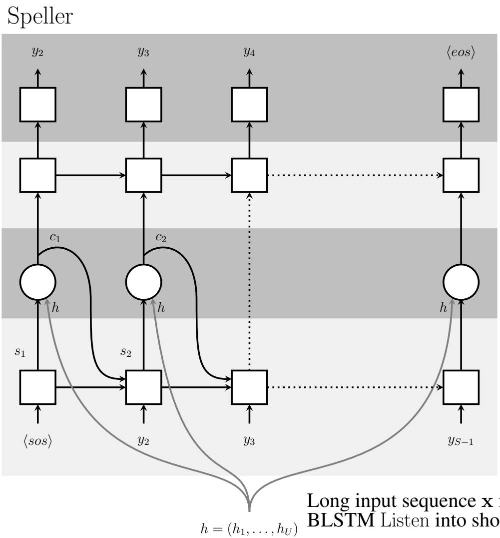
Listen Attend and Spell: Listener Module



Pyramidal BLSTM encoding input sequence **x** into high-level features **h** (†)

† Fig. from Chan et al. (2016)

Listen Attend and Spell: Speller Module†



Grapheme characters y_i are modelled by the CharacterDistribution

AttentionContext creates context vector c_i from $\mathbf{h}$ and s_i

Long input sequence $\mathbf{x}$ is encoded with the pyramidal BLSTM Listen into shorter sequence $\mathbf{h}$

†Fig. from Chan et al. (2016)

Transformers: Why not using only attention?

- Recurrent sequence-to-sequence models using encoder-decoder architecture:
  - Yield **good performances** in speech recognition
  - **Slow** (internal **recurrence** limits the training parallelization)
- To improve speed $\rightarrow$ compute speech representation with **self-attention** instead of recurrent networks (e.g., with LSTM)
- Transformers implement 2 types of attention:
  - **Self-attention** for representation
  - **Encoder-decoder attention**

Self-attention in speech

- Unlike text, speech signal is continuous:
  Need a way to discretize it

  Note: Features are actually time-discrete but in large numbers

- Differents options are proposed to handle speech features:
  - Using simple (reshape) downsampling technique (Liu et al. 2020)
  - Using CNN layers with a particular stride (Dong et al. 2018)
  - Vector quantizations (Baevski et al. 2020)
- Positional encoding (PE) needed as well
  - May cause performance degradations for longer sequences with similar acoustic attributes at different positions (Zhou et al. 2019)
  - Alternative approaches:
    - Replacing absolute PE with relative PE (Zhou et al. 2019)

Speech-Transformer (Dong et al. 2018)

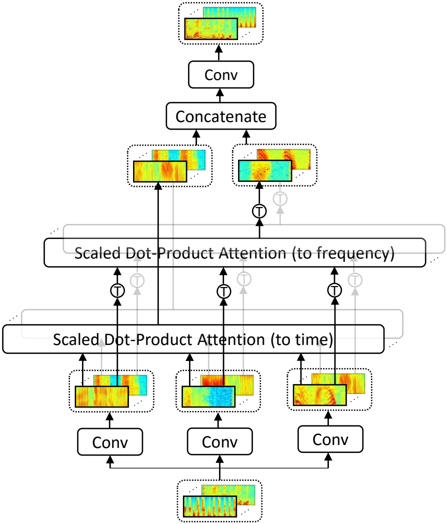
- Early transformer-based architectures in speech recognition

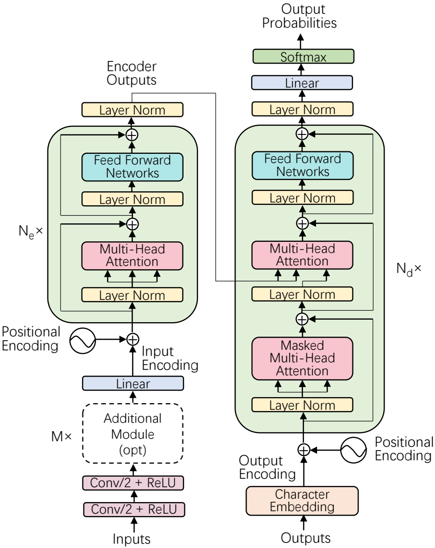
  Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition (Dong et al. 2018)

- Model relying entirely on attention mechanisms to learn the positional dependencies
  - 2D-Attention mechanism attending jointly
    (time and frequency axes)
  - Represent different *features* using different attention heads
- Minimal changes in the architecture (vs. the original Transformer)
  - Mainly: **Input embeddings through CNNs**
- Slightly lower performance than traditional SOTA models

  (proof of concept: transformer-based ASRs can work)

2D-Attention Mechanism†





†Fig. from Dong et al. (2018)

CTC Loss and Transformers

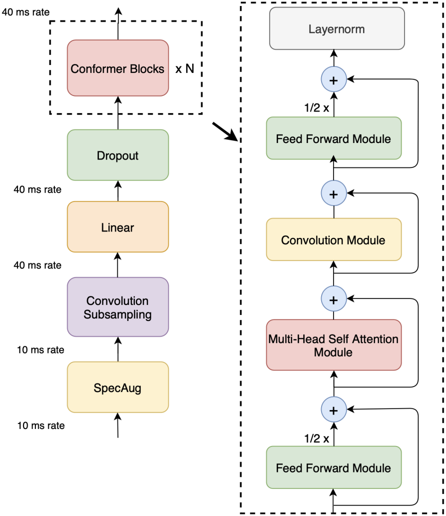
- Improvement: Integrate CTC loss into Speech-Transformer (Karita et al. 2019)
- CTC loss has several advantages:
  - Allows the alignment of audio frames to transcription characters
  - Better integration of the language model into the learning process
- Hybrid architecture combining Transformer and RNN-based ASR
- Learning curve appears to converge faster than with a pure Transformer architecture
- Evaluations:
  - WER = 4.5% on Wall Street Journal
  - WER = 11.6% on TED-LIUM

Conformer (Gulati et al. 2020) I

- Main strengths of transformer-based architectures:
  - **Fast** and **accurate**
  - Ability to capture the **global context**
- **CNNs** capture **local context** effectively
- **Combine** CNNs and transformers to model both local and global contexts
  - Add a **convolution** module **after** the **Multi-Head Attention** block
- **Conformer**:

  Convolution-augmented transformer for speech recognition
- LibriSpeech: WER = 1.9%/2.1% (with/without using a LM)

Conformer (Gulati et al. 2020) II†

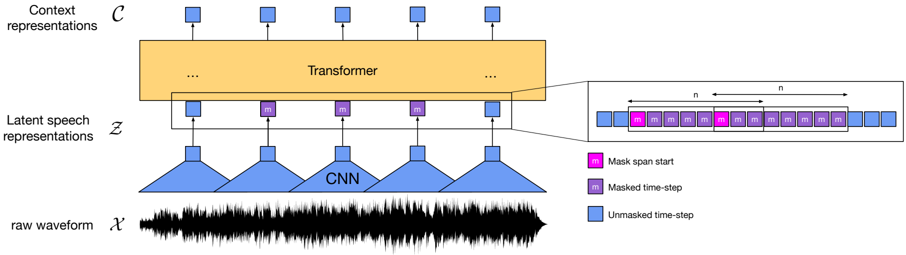


†Fig. from Gulati et al. (2020)

Self-supervised pre-training for speech

- Self-supervised learning (SSL) can be used for speech
- Like the BERT model (Devlin et al. 2018) for NLP
- BERT's task: Predict the **next sentence**
- Self-supervised pre-training can be used on large audio corpora to **learn representation without labels**
  - Helps building ASR systems with as **few** as 10 minutes of **labeled data**
  - Helps in multilingual **transfer learning**
- Popular models:
  - Wav2vec (Schneider et al. 2019) and Wav2vec 2.0 (Baevski et al. 2020)
  - Mockingjay (Liu et al. 2020)

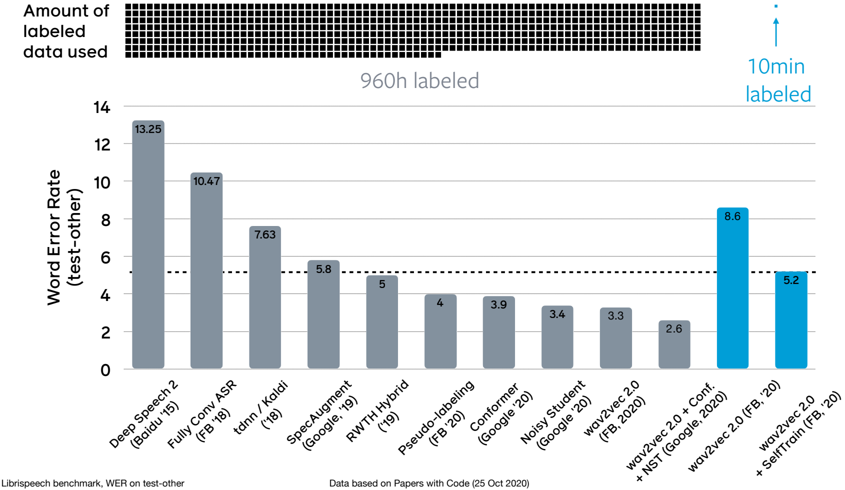
Wav2vec 2.0 (Baevski et al. 2020) I



- Fully convolutional
- Vector quantize (Jegou et al. 2010):
  Split into small segments and **cluster** them in **discrete values**
- Sample random segments (start points) for masking:†
  - Expand starting points by 10 time-steps (10 × 25ms)
  - Try to **predict** the resulting **masked segments**

†Fig. from Auli (2021)

Wav2vec 2.0 (Baevski et al. 2020) II†



†Fig. from Auli (2021)

Conclusion

- A brief overview of some chosen paper is given
- Transformers for ASR is a very active field of research
- In a short amount of time, vast improvements have been made
- Architectures are still changing but currently seem to converge toward a mixture of CNNs and Transformers
- Self-supervised learning allows to greatly improve transfer learning performances for low resources data

Bibliography I

Audhkhasi, Kartik, George Saon, Zoltán Tüske, Brian Kingsbury, and Michael Picheny. 2019. “Forget a Bit to Learn Better: Soft Forgetting for CTC-Based Automatic Speech Recognition.” In *INTERSPEECH*, 2618–22.

Auli, Michael. 2021. “Wav2vec: Self-Supervised Learning of Speech Representations.” Talk at MIT, CMU, U of Edinburgh, Spring 2021.

Baevski, Alexei, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. “Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations.” *arXiv Preprint arXiv:2006.11477*.

Chan, William, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. “Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition.” In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4960–64. IEEE.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. “Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” *arXiv Preprint arXiv:1810.04805*.

Dong, Linhao, Shuang Xu, and Bo Xu. 2018. “Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition.” In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5884–88. IEEE.

Graves, Alex, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. “Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks.” In *Proceedings of the 23rd International Conference on Machine Learning*, 369–76.

Graves, Alex, and Navdeep Jaitly. 2014. “Towards End-to-End Speech Recognition with Recurrent Neural Networks.” In *International Conference on Machine Learning*, 1764–72. PMLR.

Bibliography II

Gulati, Anmol, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, et al. 2020. "Conformer: Convolution-Augmented Transformer for Speech Recognition." *Proc. Interspeech 2020*, 5036–40.

Hinton, Geoffrey, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, et al. 2012. "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups." *IEEE Signal Processing Magazine* 29 (6): 82–97.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–80.

Jaitly, Navdeep, and Geoffrey Hinton. 2011. "Learning a Better Representation of Speech Soundwaves Using Restricted Boltzmann Machines." In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5884–87. IEEE.

Jegou, Herve, Matthijs Douze, and Cordelia Schmid. 2010. "Product Quantization for Nearest Neighbor Search." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (1): 117–28.

Karita, Shigeki, Nelson Enrique Yalta Soplin, Shinji Watanabe, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani. 2019. "Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration." *Proc. Interspeech 2019*, 1408–12.

LeCun, Yann, Yoshua Bengio, et al. 1995. "Convolutional Networks for Images, Speech, and Time Series." *The Handbook of Brain Theory and Neural Networks* 3361 (10): 1995.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. "Gradient-Based Learning Applied to Document Recognition." *Proceedings of the IEEE* 86 (11): 2278–2324.

Bibliography III

Liu, Andy T, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee. 2020. "Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders." In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6419–23. IEEE.

Mikolov, Tomas, Martin Karafiát, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. "Recurrent Neural Network Based Language Model." In *Interspeech*, 2:1045–48. 3. Makuhari.

Mohamed, Abdelrahman, Dmytro Okhonko, and Luke Zettlemoyer. 2019. "Transformers with Convolutional Context for ASR." *arXiv Preprint arXiv:1904.11660*.

Passricha, Vishal, and Rajesh Kumar Aggarwal. 2020. "A Hybrid of Deep CNN and Bidirectional LSTM for Automatic Speech Recognition." *Journal of Intelligent Systems* 29 (1): 1261–74.

Rao, Kanishka, Fuchun Peng, Haşim Sak, and Françoise Beaufays. 2015. "Grapheme-to-Phoneme Conversion Using Long Short-Term Memory Recurrent Neural Networks." In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4225–29. IEEE.

Schneider, Steffen, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. "Wav2vec: Unsupervised Pre-Training for Speech Recognition." *arXiv Preprint arXiv:1904.05862*.

Zhou, Pan, Ruchao Fan, Wei Chen, and Jia Jia. 2019. "Improving Generalization of Transformer for Speech Recognition with Parallel Schedule Sampling and Relative Positional Embedding." *arXiv Preprint arXiv:1911.00203*.