Analyzing Hidden Representations in End-to-End Automatic Speech Recognition Systems

Yonatan Belinkov, James Glass

MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA

{belinkov, glass}@mit.edu

Motivation

- Traditional Automatic Speech Recognition (ASR) systems are complex with many moving parts: acoustic model, language model, lexicon, etc.
- End-to-end ASR maps acoustics directly to text, jointly optimizing for the recognition task
- End-to-end models do not require explicit phonetic supervision (e.g. phonemes)
- **Research questions**:
  - Do end-to-end models *implicitly* learn phonetic representations ("g" in "bought")?
  - Which components capture more phonetic information?
  - Do more complicated ASR models learn better representations for phonology?

ASR Model

- DeepSpeech2 (Amodei et al. 2017):
  - Map spectrograms to characters (or blanks)
  - Stack of CNNs and RNNs

Layer	Type	Input Size	Output Size
1	cnn1	161	41x11
2	cnn2	41x11	21x11
3	rnn1	1312	1760
4	rnn2	1760	1760
5	rnn3	1760	1760
6	rnn4	1760	1760
7	rnn5	1760	1760
8	rnn6	1760	1760
9	rnn7	1760	1760
10	fc	1760	29

- CTC loss (Graves 2006)
  - Map spectrograms *x* to characters *l* by considering all possible alignments π

$$p(\mathbf{l}|\mathbf{x}) = \sum_{\pi\in\mathcal{B}^{-1}(\mathbf{l})} p(\pi|\mathbf{x}) = \sum_{\pi\in\mathcal{B}^{-1}(\mathbf{l})} \prod_{t=1}^{T} \phi_t(\mathbf{x})[\pi_t]$$

- where $\phi_t(\mathbf{x}) \in \mathbb{R}^V$ – output at time *t*

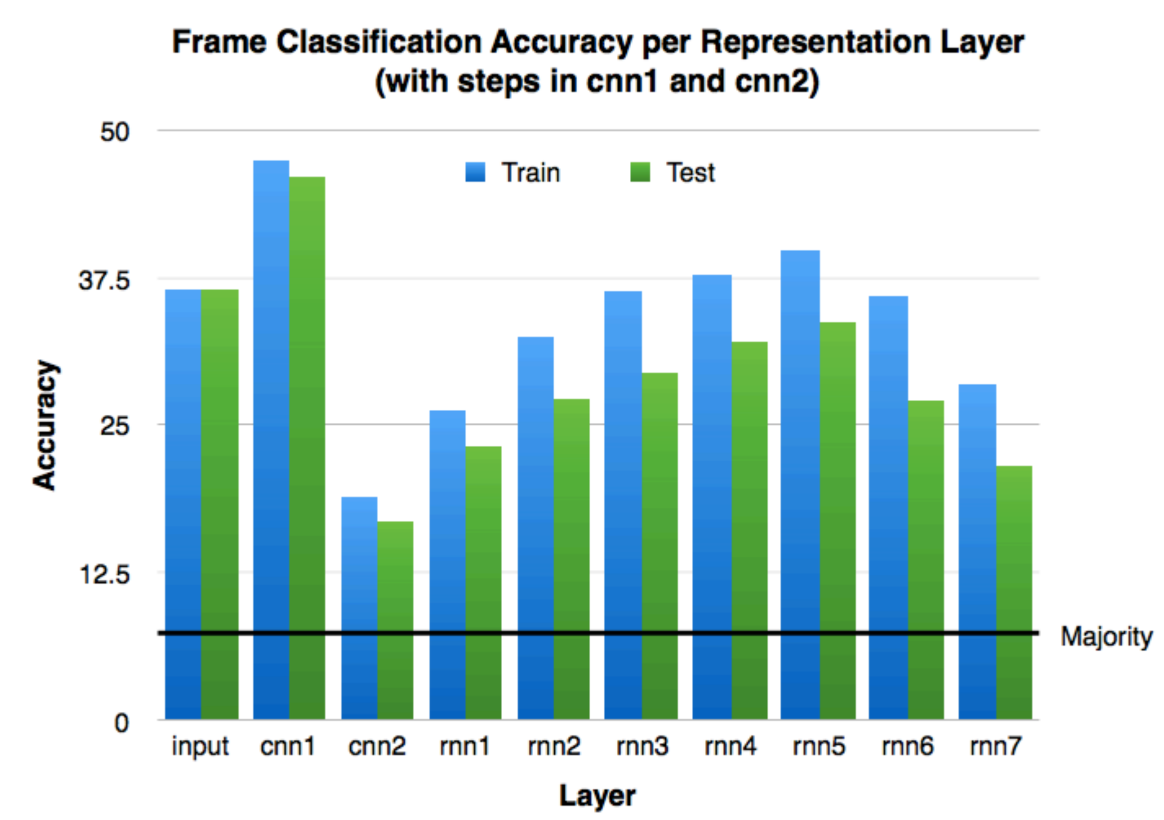
Methodology and Data

- **Methodology**
  - Train ASR model on transcribed speech
  - Extract features from the pre-trained model on a supervised dataset with phonetic segmentation
  - Train a simple classifier on a frame classification task: predict phones using the extracted features
- **Classifier**
  - One hidden layer, dropout, ReLU, softmax
  - Adam optimizer, cross-entropy loss
- **Data**
  - ASR training: LibriSpeech, 1000 hours of read speech
  - Frame classifier: TIMIT, time segmentation of phones

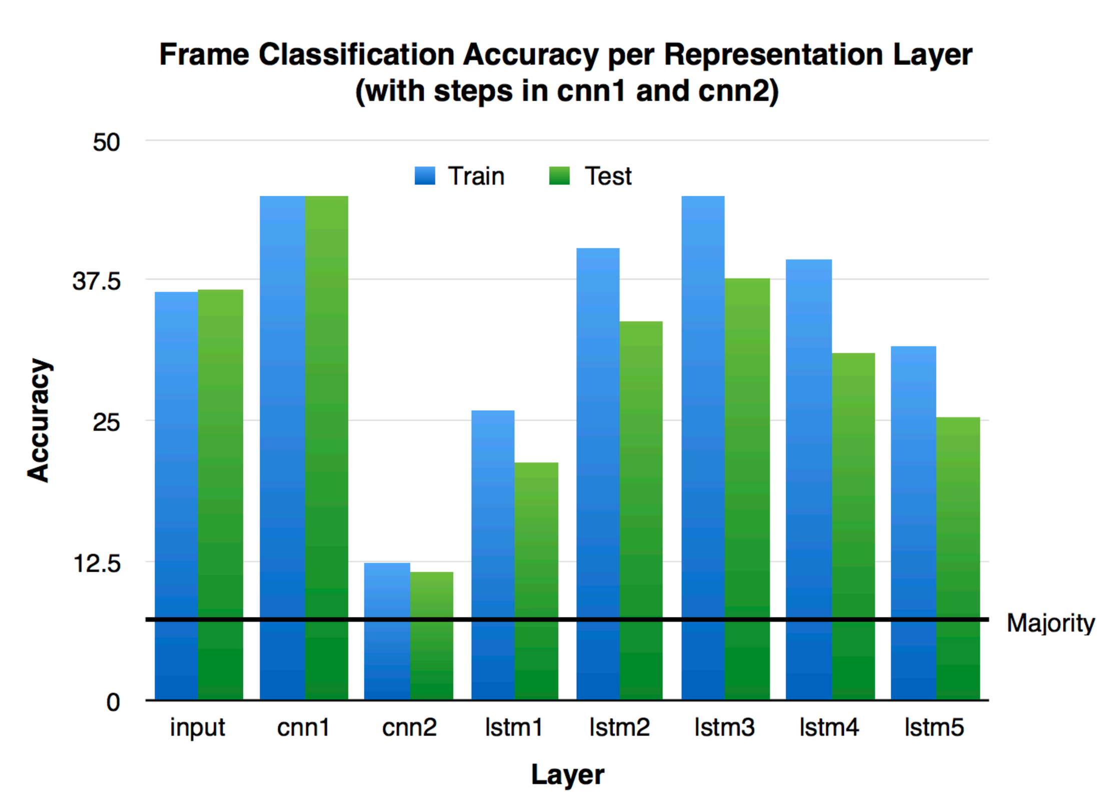
	Train	Dev	Test
Utterances	3,692	400	192
Frames	988K	108K	50K

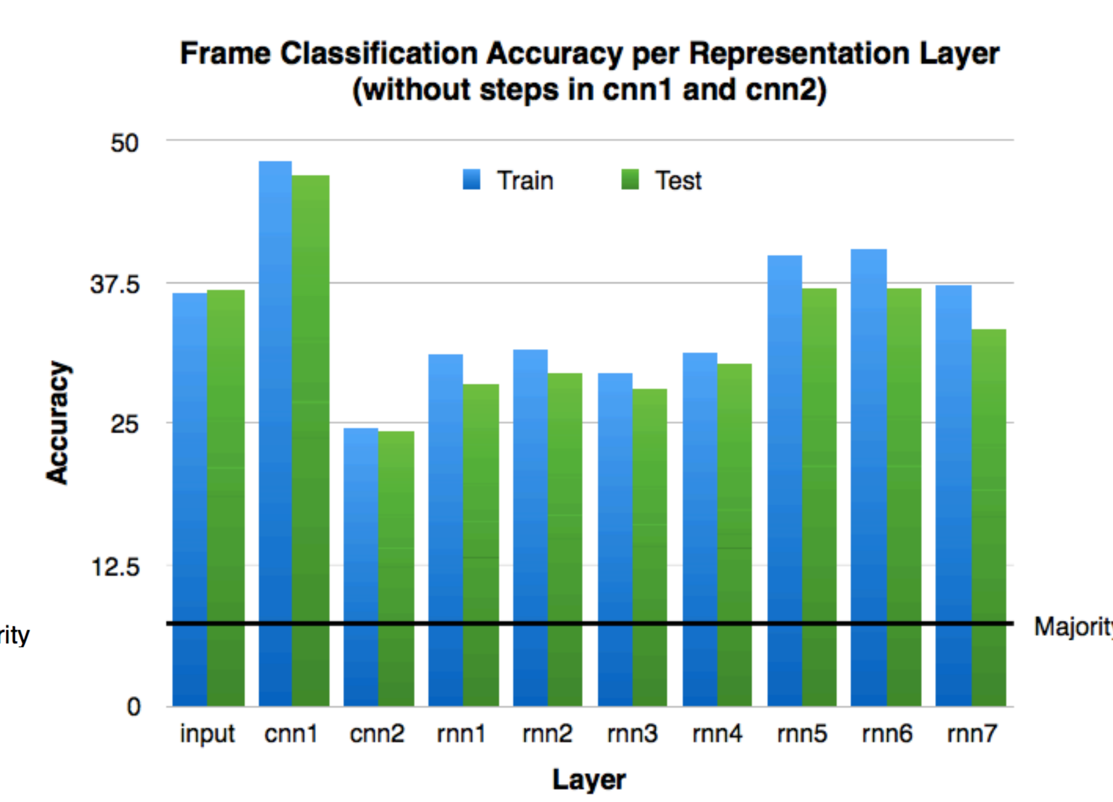
Results

- **Main results**
  - Conv1 improves the input representation, but conv2 degrades it
  - RNN layers initially improve, then drop
- ➢ Higher layers capture more global information like dependencies between characters (e.g. "bought")
- Similar trends in different configurations (layers, phone classes, input futures)



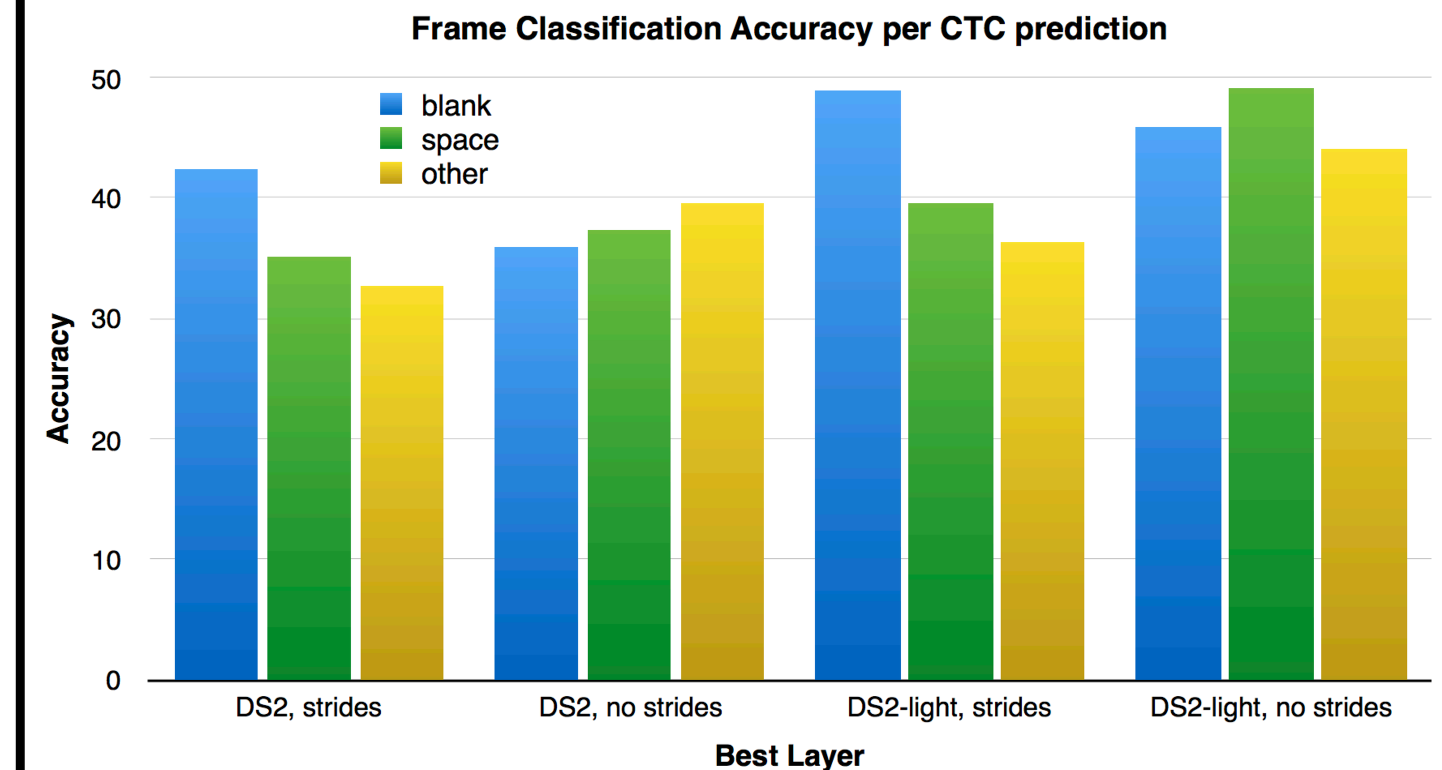
- **Model complexity**
  - LSTM layer representations are better than RNN, but the respective conv layers are worse
  - Deeper model has better WER (12 vs 15) but worse representations for phonology
- **Effect of strides**
  - Similar overall trend
  - Less spiky shape without strides, possibly thanks to higher time resolution



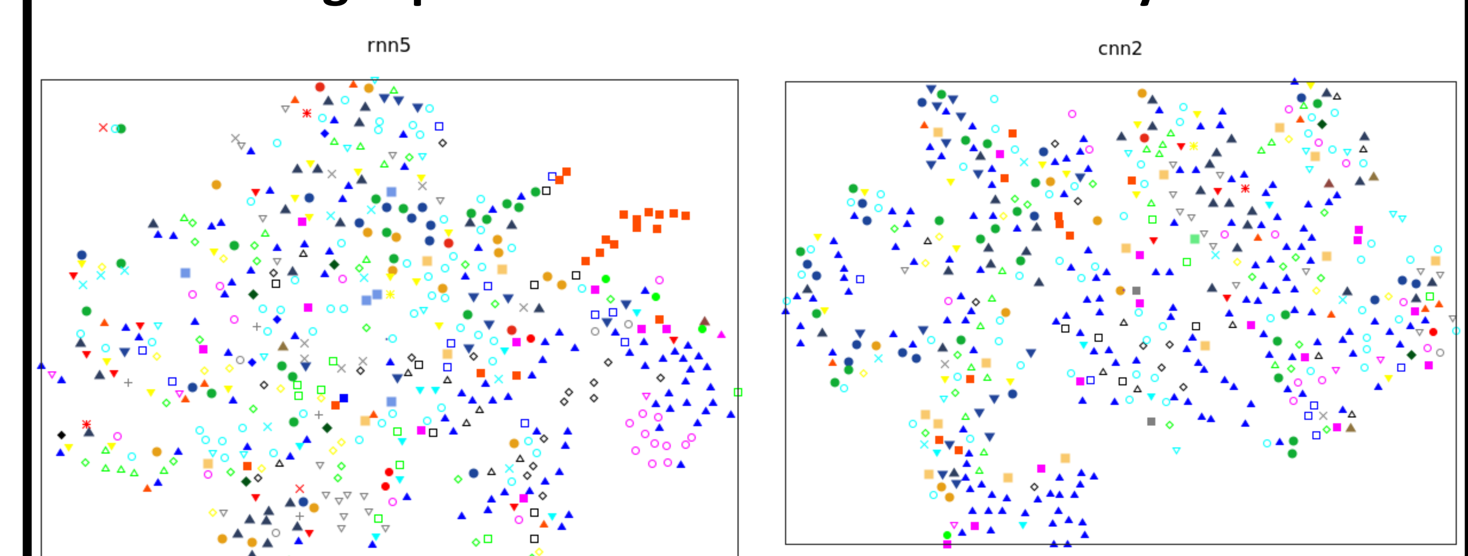


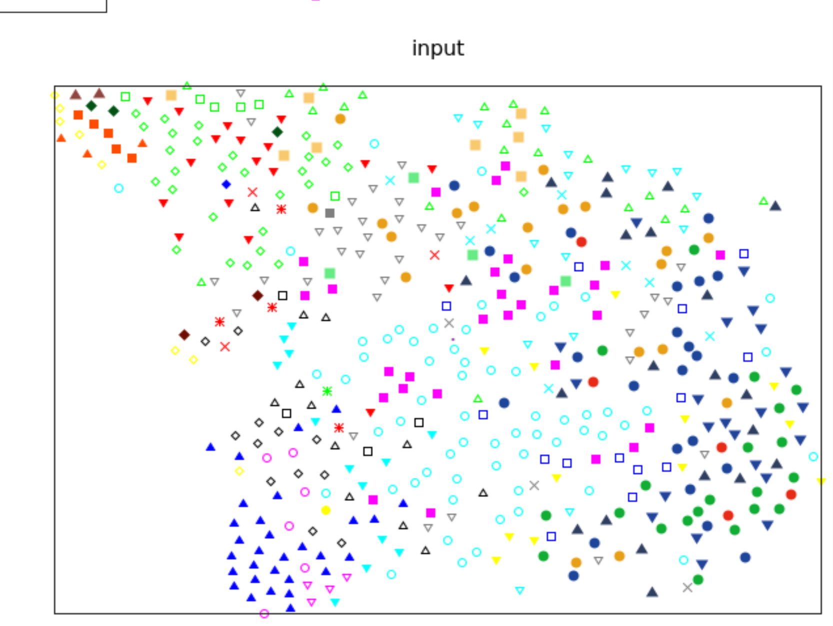
Analysis

- **Effect of blank symbols**
  - With strides, better representations at blanks
  - Without strides, better representations at non-blanks



- **Clustering representations from different layers**





- Input: good separation
- conv2: no clear groups
- rnn5: better separation

Conclusion

- End-to-end CTC models learn substantial phonetic information
- Phonetic information persists until mid-layers, but the top layers loose phonetic information
- Separability in vector space corresponds to representation quality

Acknowledgements: This work was supported by the Qatar Computing Research Institute (QCRI).