Achieving Human Parity in Conversational Speech Recognition

Speech & Dialog Research Group

AI & Research

Microsoft

Acknowledgements

ACHIEVING HUMAN PARITY IN CONVERSATIONAL SPEECH RECOGNITION

W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu and G. Zweig

Microsoft Research
Technical Report MSR-TR-2016-71

Joint work with
great collaborators!

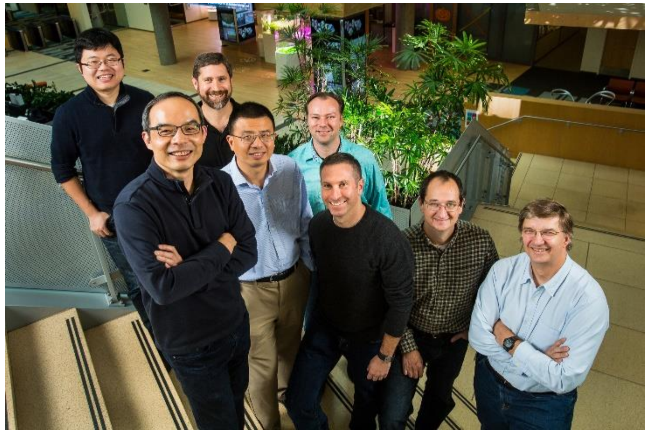
17 Oct 2016

ABSTRACT

Conversational speech recognition has served as a flagship speech recognition task since the release of the DARPA Switchboard corpus in the 1990s. In this paper, we measure the human error rate on the widely used NIST 2000 test set, and find that our latest automated system has reached human parity. The error rate of professional transcriptionists is 5.9% for the Switchboard portion of the data, in which newly acquainted pairs of people discuss an assigned topic, and 11.3%

collections of the 1990s and early 2000s provide what is to date the largest and best studied of the conversational corpora. The history of work in this area includes key contributions by institutions such as IBM [12], BBN [13], SRI [14], AT&T [15], LIMSI [16], Cambridge University [17], Microsoft [18] and numerous others.

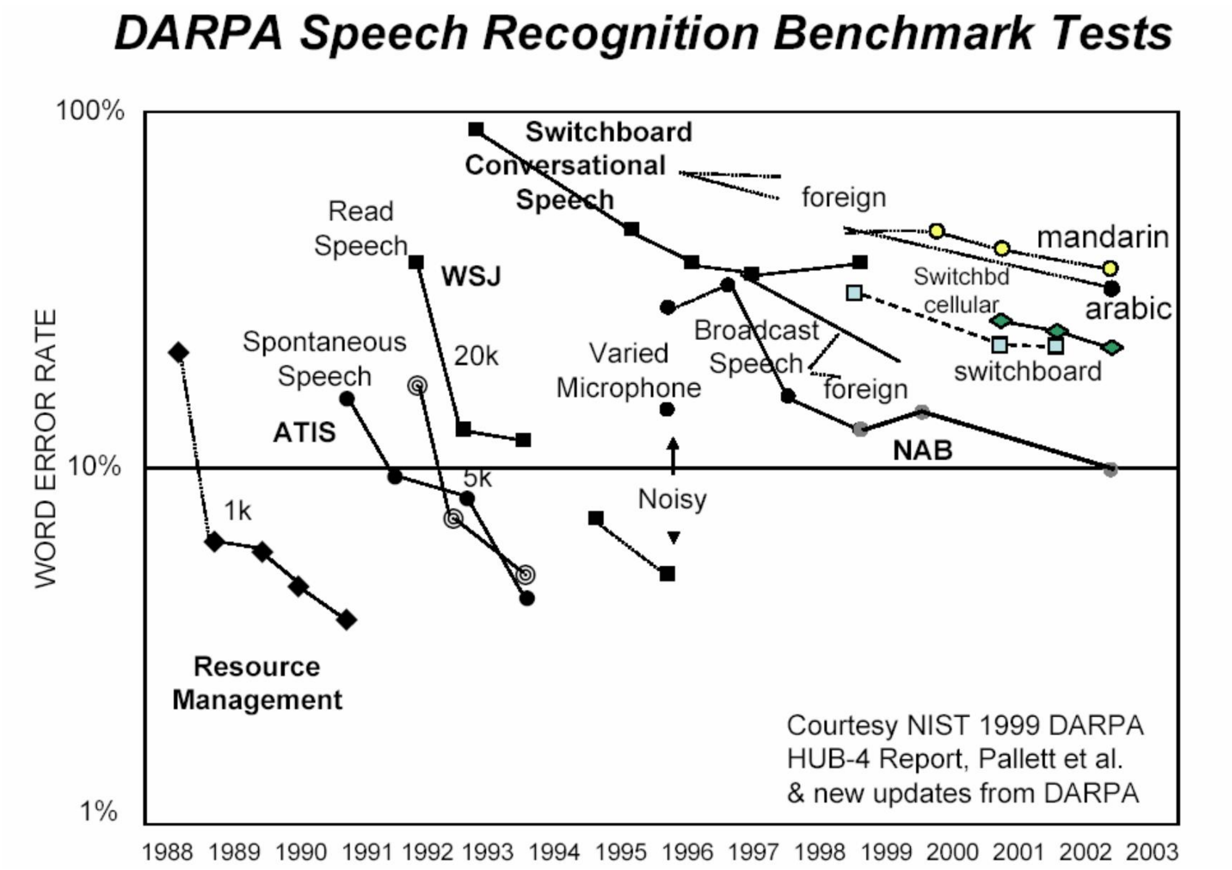
In the past, human performance on this task has been widely cited as being 4% [19]. However, the error rate estimate in [19] is attributed to a "personal communication,"



Human Parity in Conversational Speech Recognition

- What is Human Parity?
  - Humans make mistakes, too. Can ASR make fewer?
- Conversational Speech Recognition
  - Humans talking in unplanned way
  - Focus on each other, not on a computer
- The result of thirty years of progress
  - DARPA / US Government programs
  - Conversational Speech Recognition is the latest in a series of increasingly difficult tasks.

Significance: History



RM

ATIS

WSJ

SWB

CH

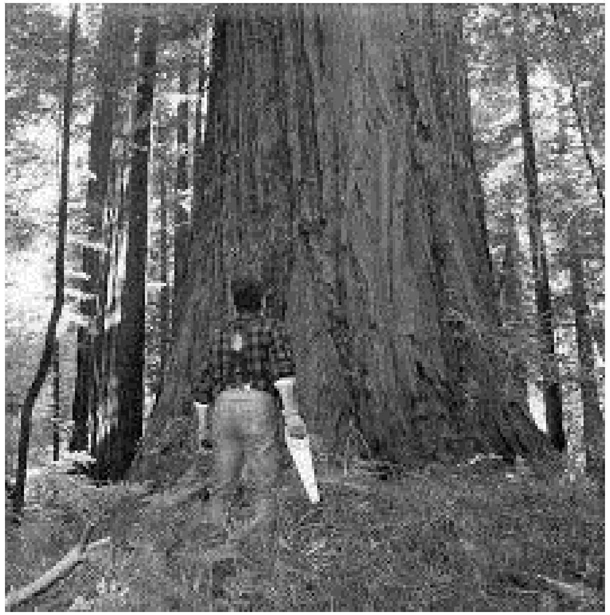
For many years, DARPA drove the field

Significance: Community

Building on accumulated knowledge of many institutions!

Significance: Technical

- The right tool for the right job
- CNNs, LSTMs!
- Building on lots of past innovations:
  - HMM modeling
  - Distributed Representations [Hinton '84]
  - Early CNNs, RNNs, TDNNs [Lang & Hinton '88, Waibel et al. '89, Robinson '91, Pineda '87]
  - Hybrid training [Renals et al. '91, Bourlard & Morgan '94]
  - Discriminative modeling
  - Speaker adaptation
  - System combination



GMMs

RNNs / CNNs

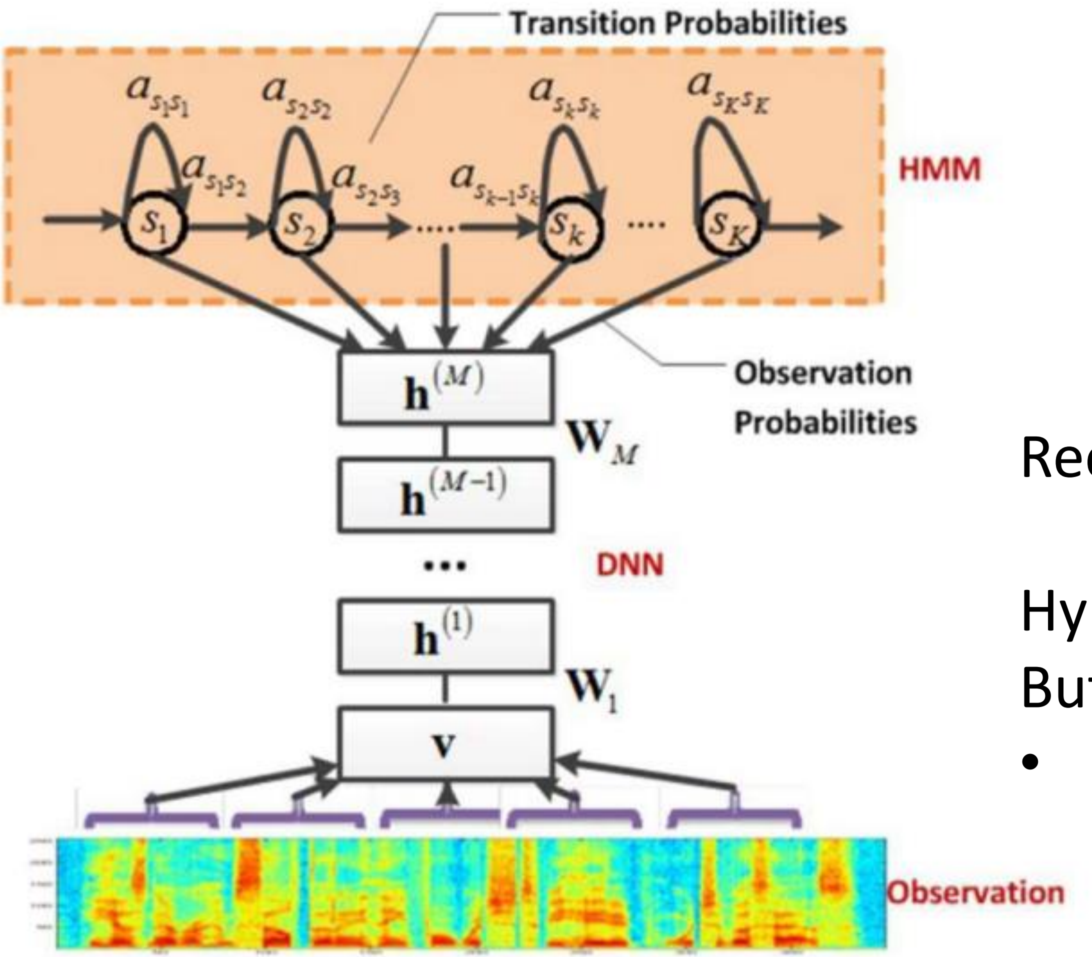
Outline

- Acoustic Modeling
- Language Modeling
- Decoding, Rescoring & System Combination
- Measuring Human Performance
- Results
- Counterpoint – Letter based CTC
- Conclusions

Outline

- Acoustic Modeling
  - Model Structures
- Language Modeling
- Decoding, Rescoring & System Combination
- Measuring Human Performance
- Results
- Counterpoint – Letter based CTC
- Conclusions

Acoustic Modeling: Hybrid HMM/DNN



[Yu et al., 2010; Dahl et al., 2011]

	CallHome	Switchboard
DNN	21.9%	13.4%

1st pass decoding

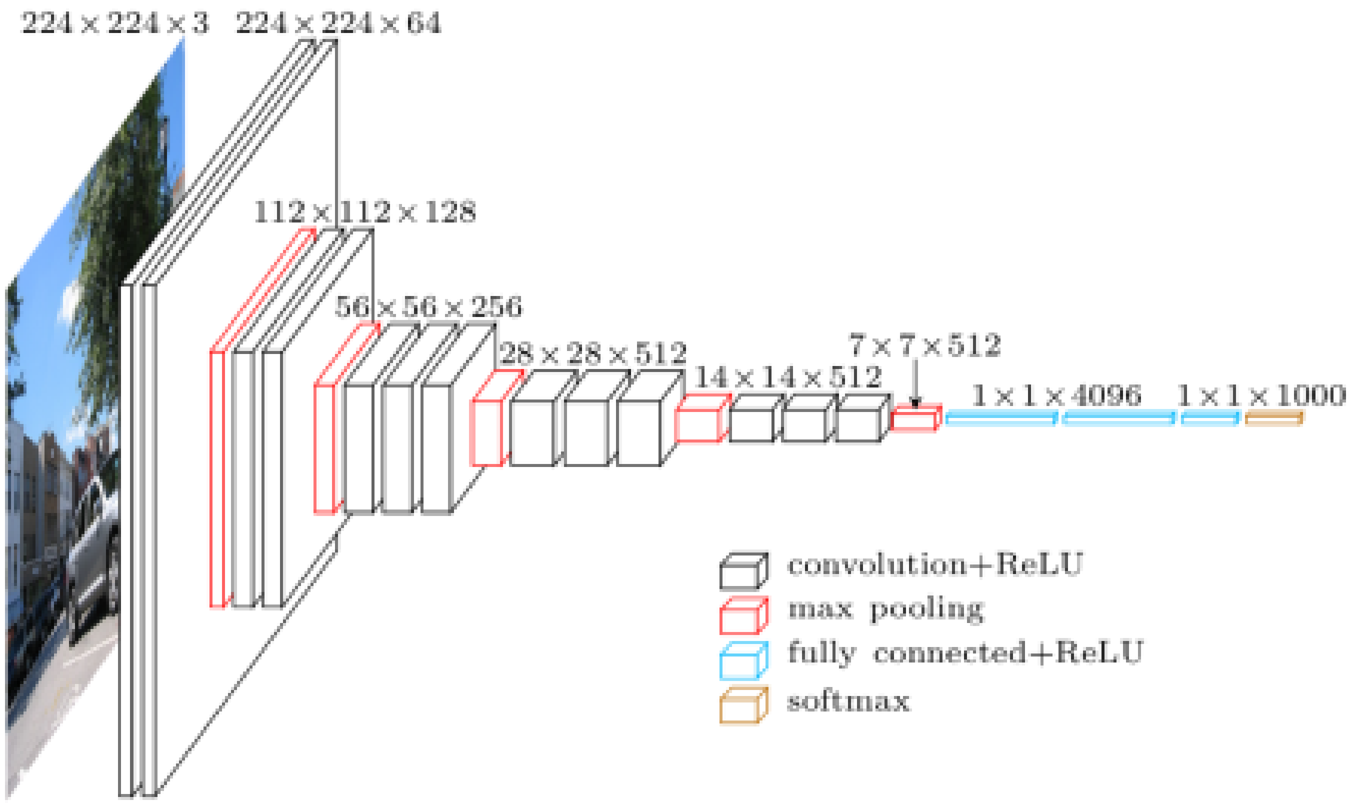
Record performance in 2011 [Seide et al.]

Hybrid HMM/NN approach standard

But DNN model now obsolete (!)

- Poor spatial/temporal invariance

Acoustic Modeling: VGG CNN



[Simonyan & Zisserman, 2014; Frossard 2016, Saon et al., 2016, Krizhevsky et al., 2012]

Adapted from image processing

Robust to temporal and frequency shifts

Acoustic Modeling: ResNet

Add a non-linear offset to linear transformation of features

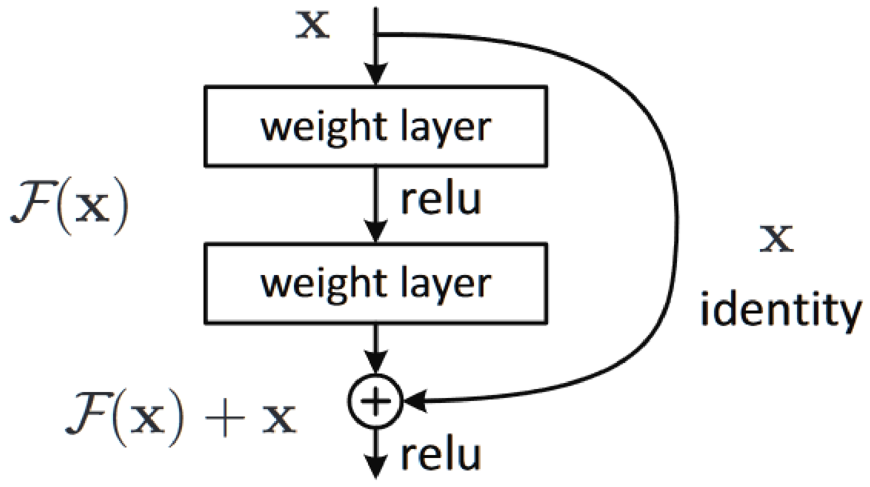
Similar to fMPE in Povey et al., 2005

See also Ghahremani & Droppo, 2016

Our best single model after rescoring

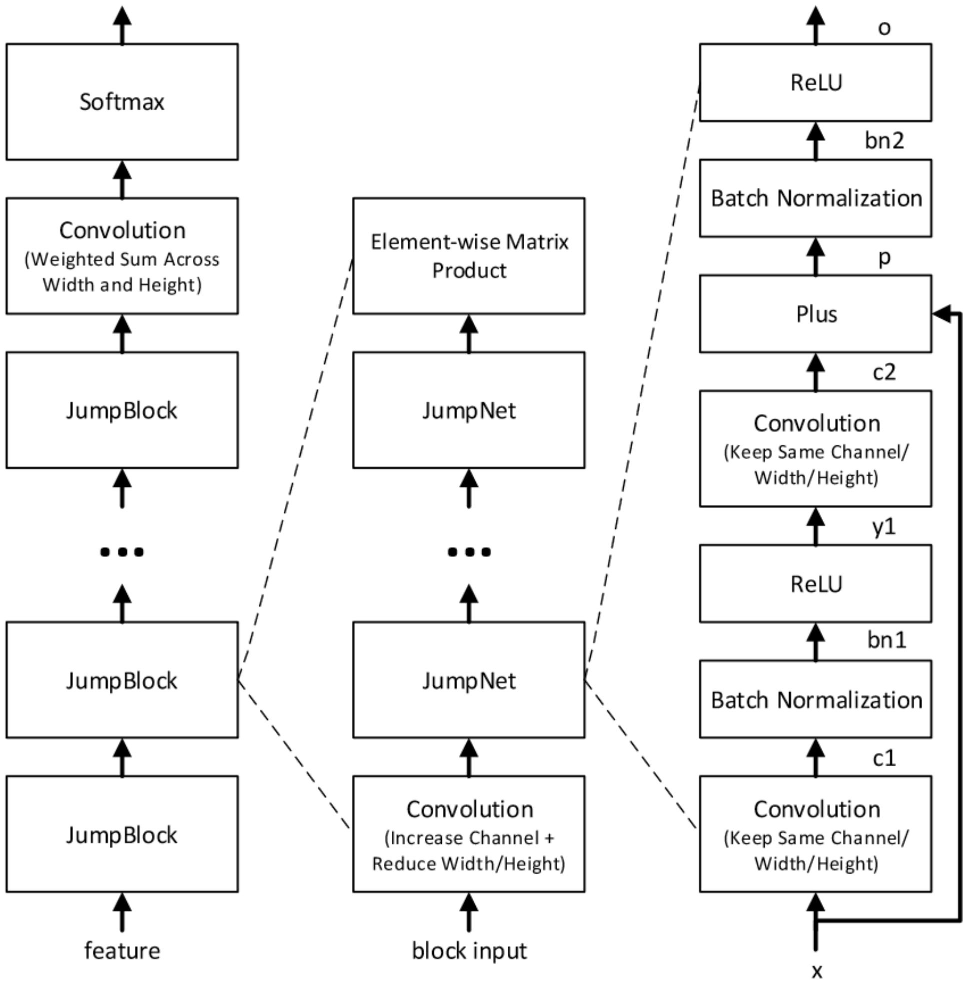
	CallHome	Switchboard
DNN	21.9%	13.4%
ResNet	17.3%	11.1%

1st pass decoding



[He et al., 2015]

Acoustic Modeling: LACE CNN



[Yu et al., 2016]

	CallHome	Switchboard
DNN	21.9%	13.4%
ResNet	17.3%	11.1%
LACE	16.9%	10.4%

1st pass decoding

Combines batch normalization, Resnet jumps, and attention masks in CNN

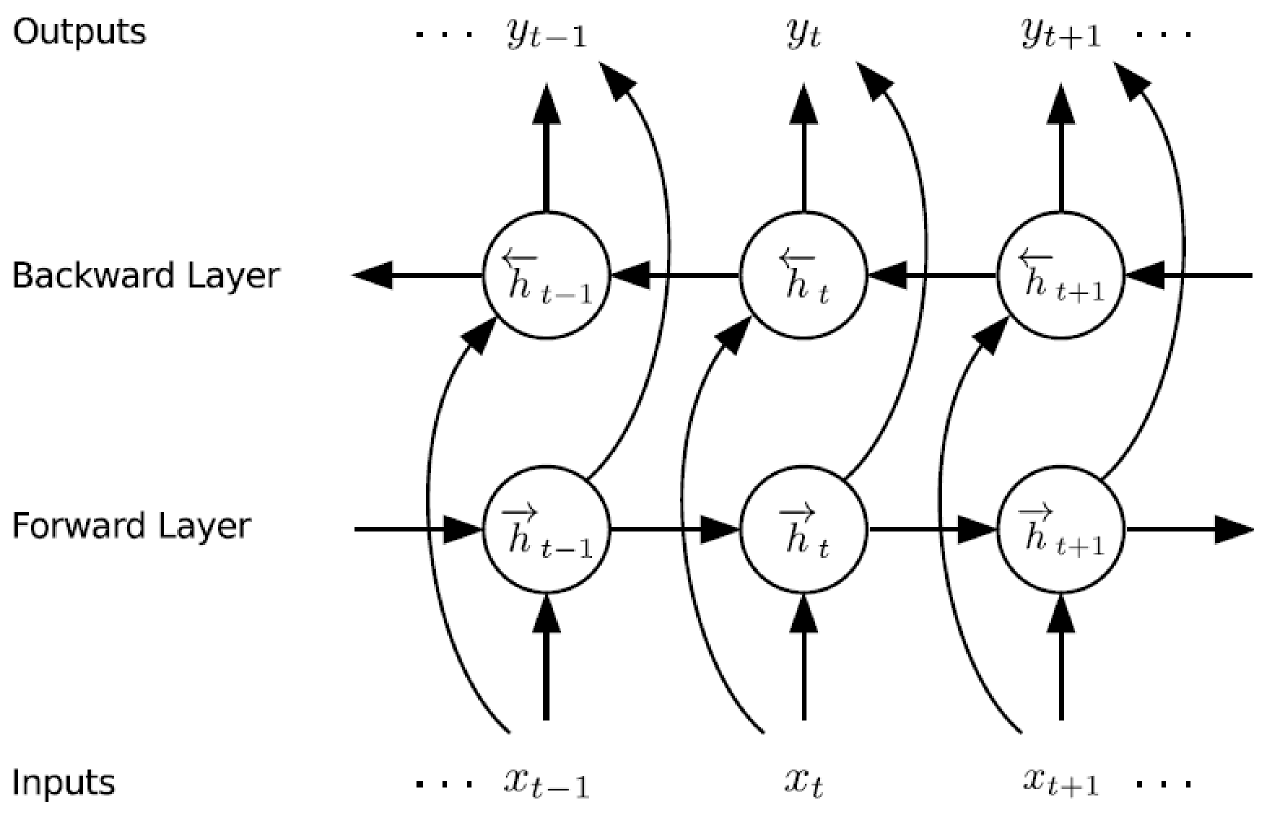
Tied for 2nd best single model after rescoring

CNN Comparison

VGG Net (85M Parameters)	Residual-Net (38M Parameters)	LACE (65M Parameters)
14 weight layers	49 weight layers	22 weight layers
40x41 input	40x41 input	40x61 input
3 – conv 3x3, 96	3 – [conv 1x1, 64 conv 3x3, 64 conv 1x1, 256]	5 – conv 3x3, 128
Max pool	4 – [conv 1x1, 128 conv 3x3, 128 conv 1x1, 512]	5 – conv 3x3, 256
4 – conv 3x3, 192	6 – [conv 1x1, 256 conv 3x3, 256 conv 1x1, 1024]	5 – conv 3x3, 512
Max pool	3 – [conv 1x1, 512 conv 3x3, 512 conv 1x1, 2048]	5 – conv 3x3, 1024
4 – conv 3x3, 384	Average pool	1 – conv 3x4, 1
Max pool	Softmax (9000)	Softmax (9000)
2 – FC – 4096		
Softmax (9000)		

Very deep
Many parameters
Small convolution patterns
Processing ~ ½ second per window

Acoustic Modeling: Bidirectional LSTMs



[Graves & Jaitly '14]

	CallHome	Switchboard
DNN	21.9%	13.4%
ResNet	17.3%	11.1%
LACE	16.9%	10.4%
BLSTM	17.3%	10.3%

Stable form of recurrent neural net

Robust to temporal shifts

Tied for 2nd best single model

[Hochreiter & Schmidhuber, 1997,
Graves & Schmidhuber, 2005; Sak et al., 2014]

Outline

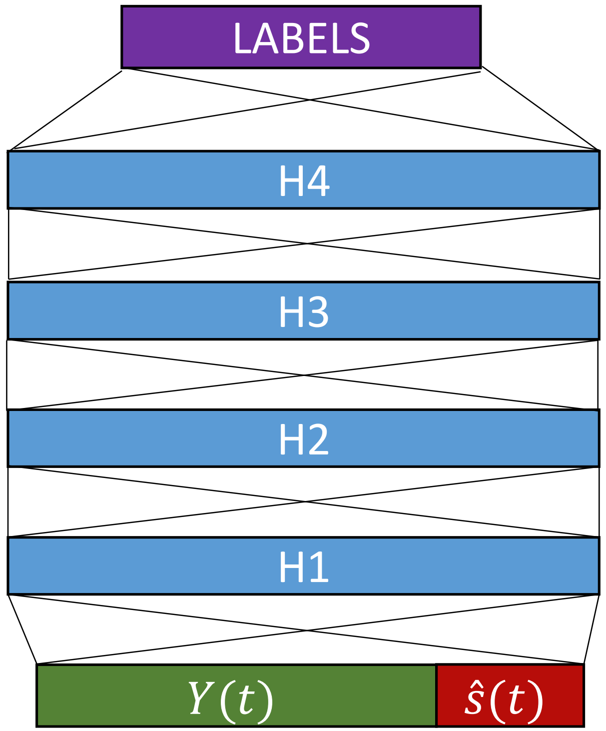
- Acoustic Modeling
    - Training Techniques
- Language Modeling
- Decoding, Rescoring & System Combination
- Measuring Human Performance
- Results
- Counterpoint – Letter based CTC
- Conclusions

I-vector Adaptation

5-10% relative improvement for Switchboard

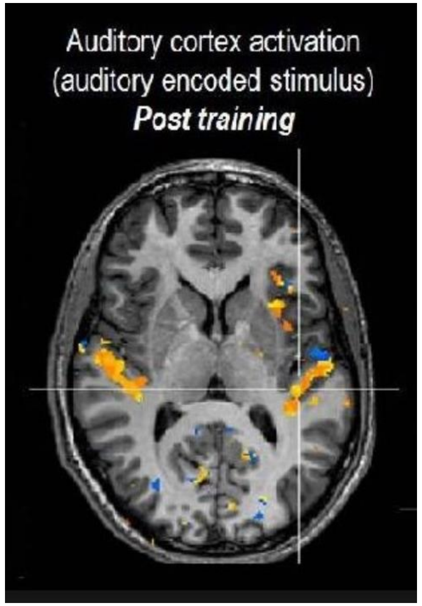
Configuration	ResNet		LACE		BLSTM	
	CH	SWB	CH	SWB	CH	SWB
Baseline	17.5	11.1	16.9	10.4	17.3	10.3
i-vector	16.6	10.0	16.4	9.3	17.6	9.9

I-vectors provide a fixed-length representation of a speaker's voice characteristics.



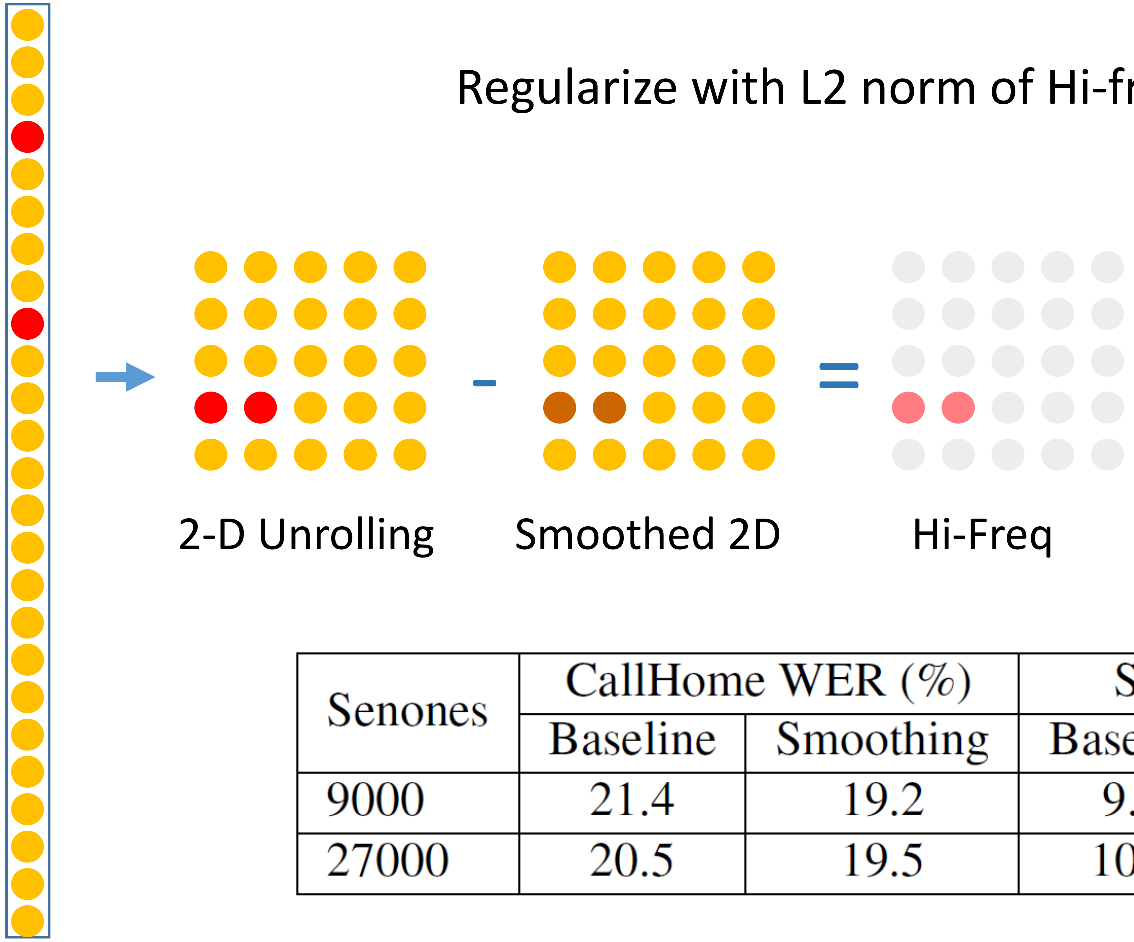
[Dehak et al. 2011; Saon et al., 2013]

Spatial Regularization



[Droppo et al. in progress]

Regularize with L2 norm of Hi-frequency residual



Senones	CallHome WER (%)		SWB WER (%)	
	Baseline	Smoothing	Baseline	Smoothing
9000	21.4	19.2	9.9	9.3
27000	20.5	19.5	10.6	9.2

5-10% relative improvement for BLSTM

Lattice Free MMI

$$\arg\max_{\Theta} \sum_{w,a \in \text{Data}} \log \frac{P(w,a;\Theta)}{P(w)P(a;\Theta)}$$

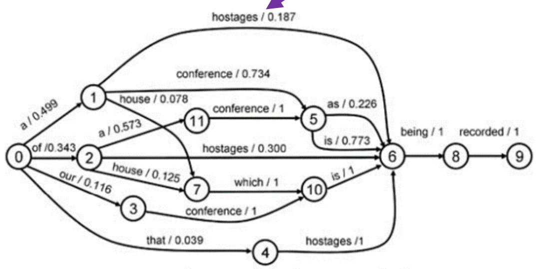
$$= \arg\max_{\Theta} \sum_{w,a \in \text{Data}} \log \frac{P(a \mid w;\Theta)}{P(a;\Theta)}$$

$$= \arg\max_{\Theta} \sum_{w,a \in \text{Data}} \log \frac{P(a \mid w;\Theta)}{\sum_{w'} P(w')P(a \mid w';\Theta)}$$

- Simple brute force MMI
- Avoids need to generate lattices
- Alignments always current

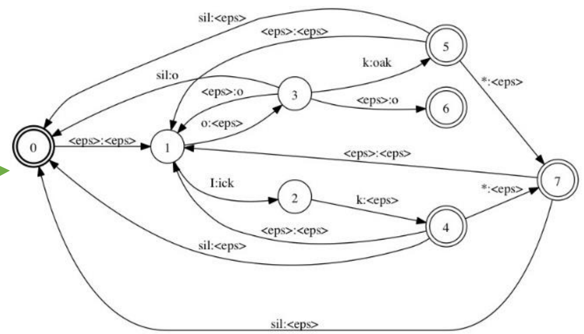
[Chen et al., 2006, McDermott et al., 2914, Povey et al., 2016]

Traditionally approximated by word sequences in lattice (DAG)

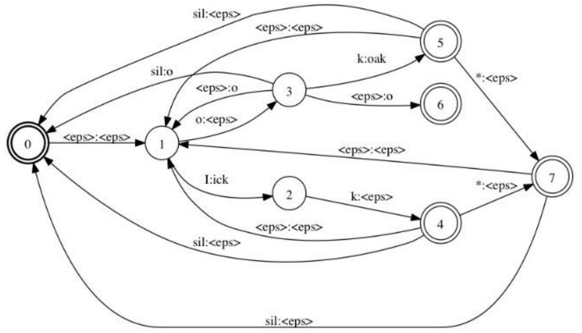


a conference is being recorded

Instead LFMMI uses all possible word sequences in cyclic FSA



Denominator GPU computation



- Represent FSA of all possible state sequences as a sparse transition matrix A
- Implement exact alpha beta computations

$$\alpha_t = \left(\mathbf{A}\alpha_{t-1}\right)\cdot o_t$$

$$\beta_t = \mathbf{A}^T\left(\beta_{t+1}\cdot o_{t+1}\right)$$

- Execute in straight “for” loops on GPU with **cusparseDcsrmv** and **cublasDdgmm**
- Beautifully simple

LFMMI Improvements

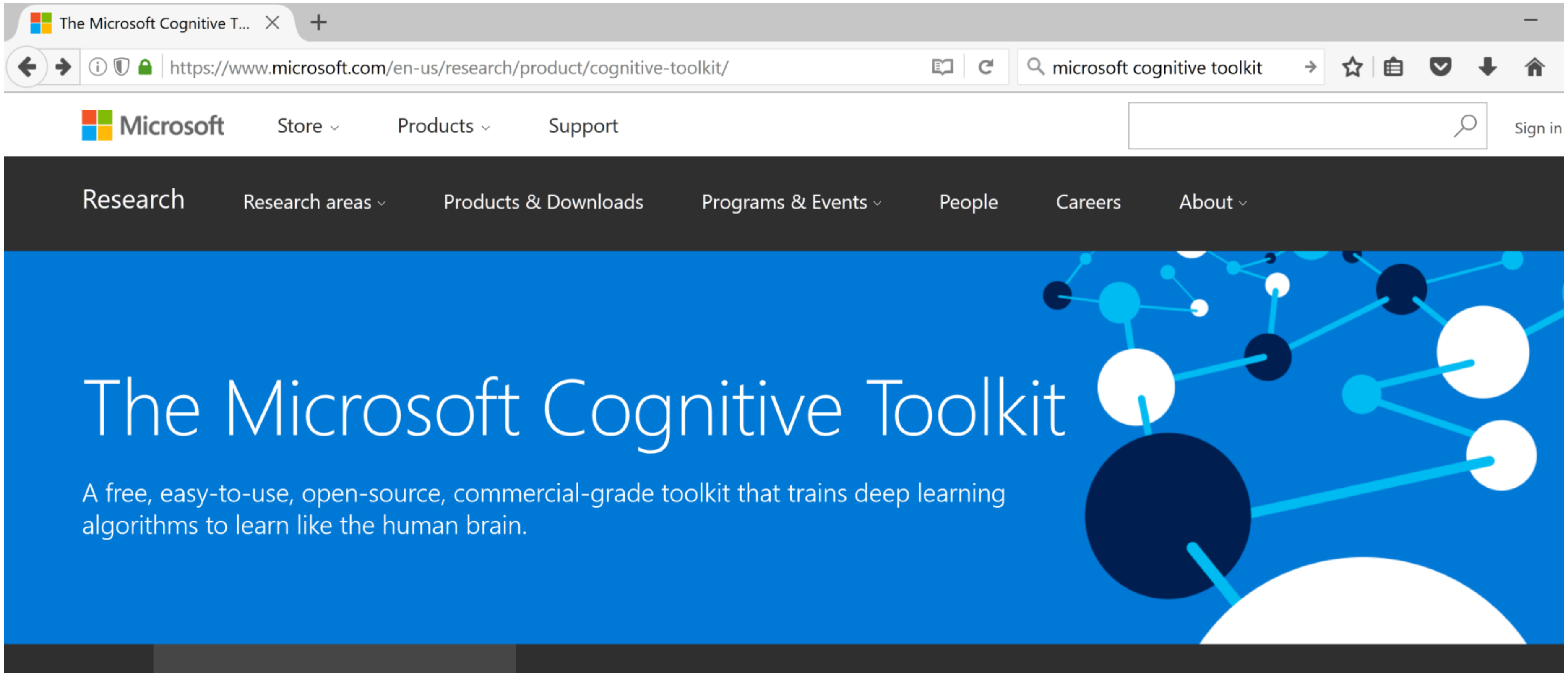
Configuration	ResNet		LACE		BLSTM	
	CH	SWB	CH	SWB	CH	SWB
Baseline	17.5	11.1	16.9	10.4	17.3	10.3
i-vector	16.6	10.0	16.4	9.3	17.6	9.9
i-vector+LFMMI	15.2	8.6	16.2	8.5	16.3	8.9

8-14% relative improvement on SWB

- Denominator LM graph has 52k states and 215k transitions
- GPU-side alpha-beta computation is 0.18xRT exclusive of NN evaluation

Cognitive Toolkit (CNTK) Training

- Flexible
- Multi-GPU
- Multi-Server
- 1-bit SGD
- All AM training
- Best LM training

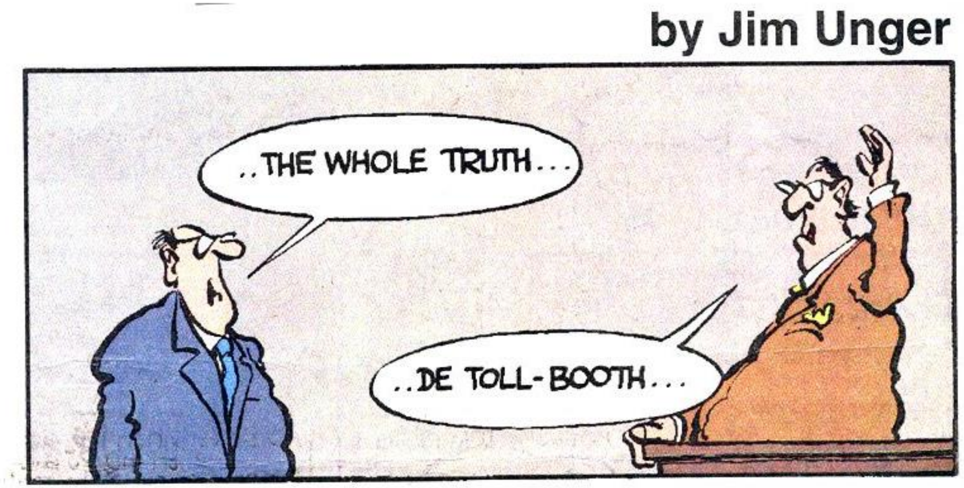


Outline

- Acoustic Modeling
- Language Modeling
- Decoding, Rescoring & System Combination
- Measuring Human Performance
- Results
- Counterpoint – Letter based CTC
- Conclusions

Language Models

- 1st Pass n-gram:
  - SRI-LM, 30k vocab, 16M n-grams
- Rescoring n-gram:
  - SRI-LM, 145M n-grams
- RNN LM
  - CUED Toolkit, two 1000 unit layers
  - Relu activations, NCE training
- LSTM LM
  - Cognitive Toolkit (CNTK), three 1000 unit layers
  - Letter trigram input, no NCE

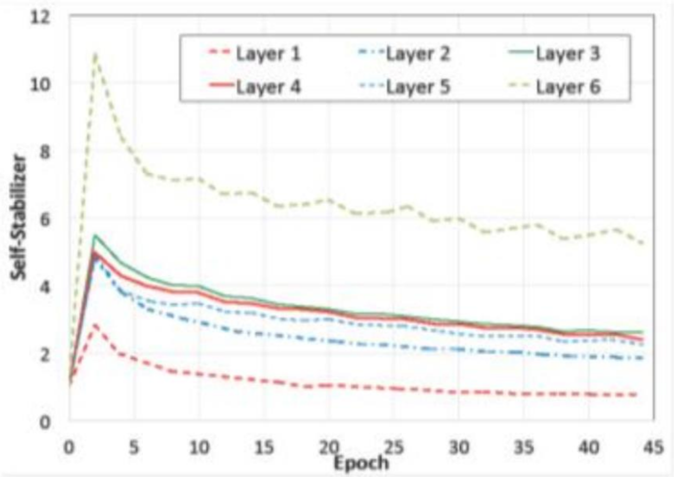


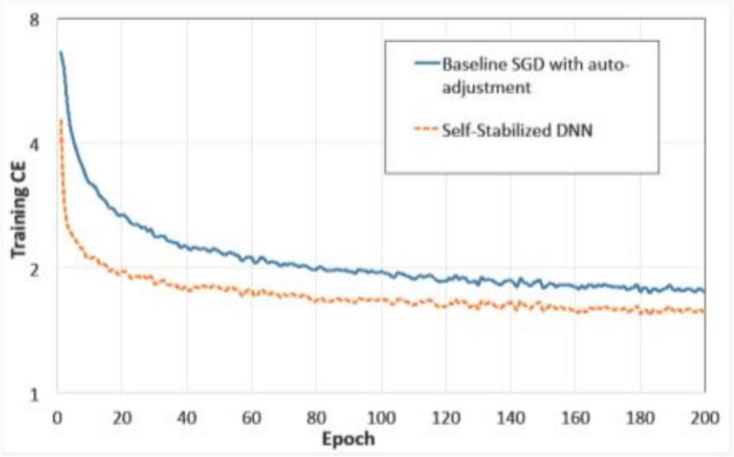
LM Training Trick: Self-stabilization

- Learn an overall scaling function for each layer

$$\mathbf{y} = \mathbf{W}\mathbf{x} \quad \text{becomes:}$$
$$\mathbf{y} = (\beta\,\mathbf{W})\mathbf{x}$$

Applied to the LSTM networks, between layers.





[Ghahremani & Droppo, 2016]

Language Model Perplexities

Language model	PPL
Ngram: 4gram baseline (145M ngrams)	75.5
RNN: 2 layers + word input	59.8
LSTM: word input in forward direction	54.4
LSTM: word input in backward direction	53.4
LSTM: letter trigram input in forward direction	52.1
LSTM: letter trigram input in backward direction	52.0

Perplexities on the 1997 eval set

LSTM beats RNN

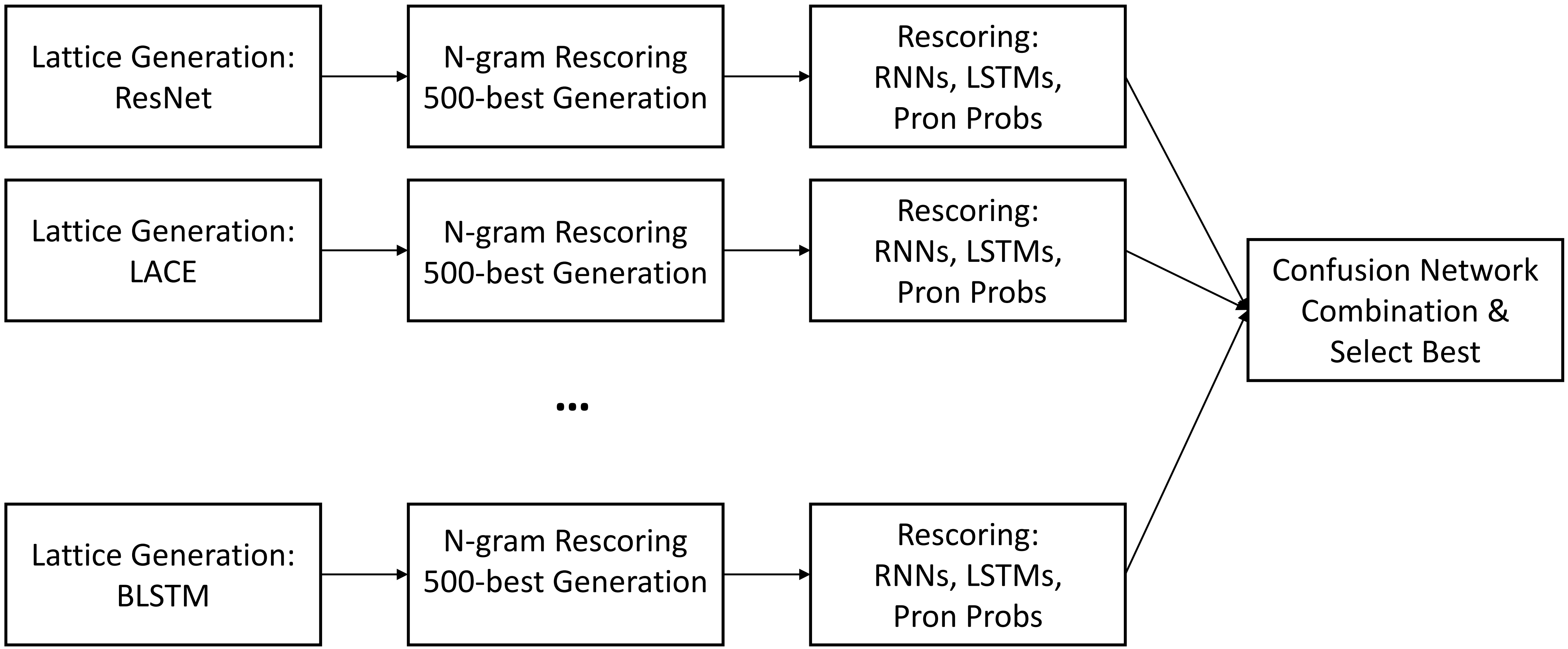
Letter trigram input slightly better than word input

Note both forward and backward running models

Outline

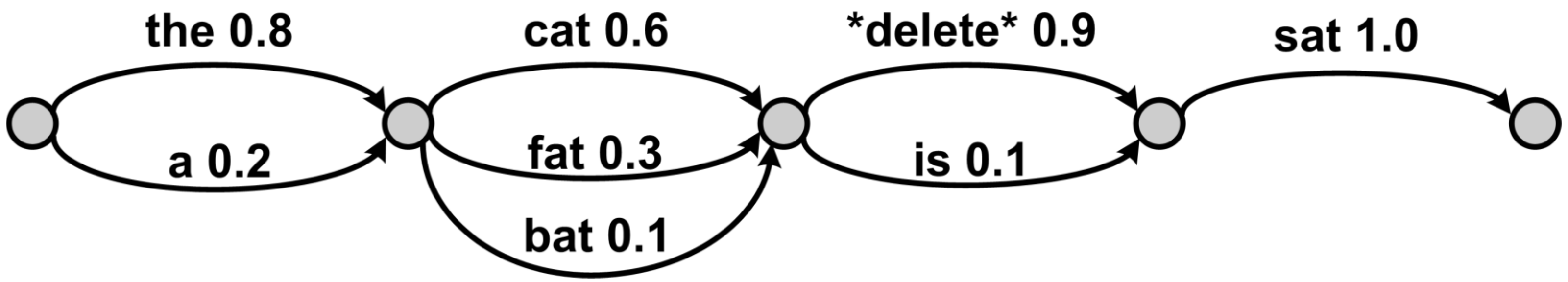
- Acoustic Modeling
- Language Modeling
- Decoding, Rescoring & System Combination
- Measuring Human Performance
- Results
- Counterpoint – Letter based CTC
- Conclusions

Overall Process



Greedy System Combination

- Make confusion network from best single system
- Repeat:
  - Compute error rate on development data for each possible system addition
  - Add the system



Rescoring Performance

Language model	PPL	WER
4-gram LM (baseline)	75.5	8.6
+ RNN-LM	59.8	7.4
+ LSTM-LM	51.4	6.9
+ 2-LSTM-LM interpolation	50.5	6.8
+ 2FW & 2 BW	-	6.6

One LSTM ~ 0.5% better than one RNN.

Multiple LSTMs provide further 0.3%

ResNet CNN Acoustic Model (no combination)

Outline

- Acoustic Modeling
- Language Modeling
- Decoding, Rescoring & System Combination
- Measuring Human Performance
- Results
- Counterpoint – Letter based CTC
- Conclusions

A First Try

- The 4% rumor

[Lippman, 1997]

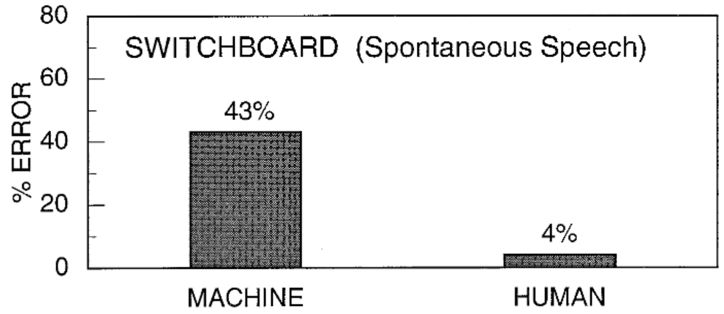


Fig. 7. Word error rates for humans and a high-performance HMM recognizer on phrases extracted from spontaneous telephone conversations in the Switchboard speech corpus (Liu et al., 1996; Martin, 1996).

1996. Speech recognition on Mandarin Call Home: A large-vocabulary, conversational, and telephone speech corpus. Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., pp. 157–160.

A. Martin, 1996. Personal communication.

Miller, G.A., 1962. Decision units in the perception of speech. Institute of Radio Engineers Transactions on Information Theory 8, 81–83.

Another Attempt

Language	Genre	Careful Transcription WDR	Quick (Rich) Transcription WDR
English	CTS	4.1-4.5%	9.63% (5 pairs)
	Meeting	-	6.23% (4 pairs)
	Interview	n/a	3.84% (22 pairs)
	BN	1.3%	3.5% (6 pairs)
	BC	n/a	6.3% (6 pairs)

[Glenn et al., 2010]

Significant variability.

Note the bulk of the training data was "quick transcribed."

Getting a Positive ID on Actual Test Data

- Skype Translator has a weekly transcription contract
  - Quality control, training, etc.
- Transcription followed by a second checking pass
- One week, we added eval 2000 to the pile…

The Results

- Switchoard: **5.9%** error rate
- CallHome: **11.3%** error rate

- SWB in the 4.1% - 9.6% range expected

- CH is *difficult for both people and machines*
  - High ASR error not just because of mismatched conditions

Language	Genre	Careful Transcription WDR	Quick (Rich) Transcription WDR
English	CTS	4.1-4.5%	9.63% (5 pairs)
	Meeting	-	6.23% (4 pairs)
	Interview	n/a	3.84% (22 pairs)
	BN	1.3%	3.5% (6 pairs)
	BC	n/a	6.3% (6 pairs)

Outline

- Acoustic Modeling
- Language Modeling
- Decoding, Rescoring & System Combination
- Measuring Human Performance
- Results
- Counterpoint – Letter based CTC
- Conclusions

The Bottom Line & Comparisons

Model	N-gram LM		RNN-LM		LSTM-LM	
	CH	SWB	CH	SWB	CH	SWB
ResNet	14.8	8.6	13.2	6.9	12.5	6.6
LACE	14.8	8.3	13.5	7.1	12.7	6.7
BLSTM (27k, spatial smoothing)	14.9	8.3	13.7	7.0	13.0	6.7
Final ASR System	13.3	7.4	12.0	6.2	**11.1**	**5.9**
Human Performance	-	-	-	-	11.3	5.9

Single CNN does remarkably well

Parity with professional transcribers

Model	N-gram LM		NN LM	
	CH	SWB	CH	SWB
Saon et al. [51] LSTM	15.1	9.0	-	-
Povey et al. [54] LSTM	15.3	8.5	-	-
Saon et al. [51] Combination	13.7	7.6	12.2	6.6

Best previous number

Runtimes

	DNN	BLSTM	ResNet	LACE
AM Training, GPU	0.012	0.022	0.60	0.23
AM eval, GPU	0.0064	0.0081	0.15	0.081
AM eval, CPU	0.052	NA	11.7	8.47
Decoding, GPU	1.04	1.40	1.19	1.38

GPU 10 to 100x faster than CPU

(Multiples of real-time, smaller is better)

AM Training: Forward, Backward + Update computations
AM eval: Forward probability computation only
Decoding: Mixed GPU/CPU, complete decoding time with open beams
Titan X GPU & Intel Xeon E5-2620 v3 @2.4GhZ, 12 cores
All times are xRT (fraction of real-time required) on Titan X GPU

Error Analysis

Substitutions (~21k words in each test set)

CH		SWB	
ASR	**Human**	**ASR**	**Human**
45: (%hesitation) / %bcack	12: a / the	29: (%hesitation) / %bcack	12: (%hesitation) / hmm
12: was / is	10: (%hesitation) / a	9: (%hesitation) / oh	10: (%hesitation) / oh
9: (%hesitation) / a	10: was / is	9: was / is	9: was / is
8: (%hesitation) / oh	7: (%hesitation) / hmm	8: and / in	8: (%hesitation) / a
8: a / the	7: bentsy / bensi	6: (%hesitation) / i	5: in / and
7: and / in	7: is / was	6: in / and	4: (%hesitation) / %bcack
7: it / that	6: could / can	5: (%hesitation) / a	4: and / in
6: in / and	6: well / oh	5: (%hesitation) / yeah	4: is / was

“ums” and “uh-hums” most frequent mistakes
– but most errors are in the long tail

Error Analysis

Deletions

CH		SWB	
ASR	**Human**	**ASR**	**Human**
44: i	73: i	31: it	34: i
33: it	59: and	26: i	30: and
29: a	48: it	19: a	29: it
29: and	47: is	17: that	22: a
25: is	45: the	15: you	22: that
19: he	41: %bcack	13: and	22: you
18: are	37: a	12: have	17: the
17: oh	33: you	12: oh	17: to

Insertions

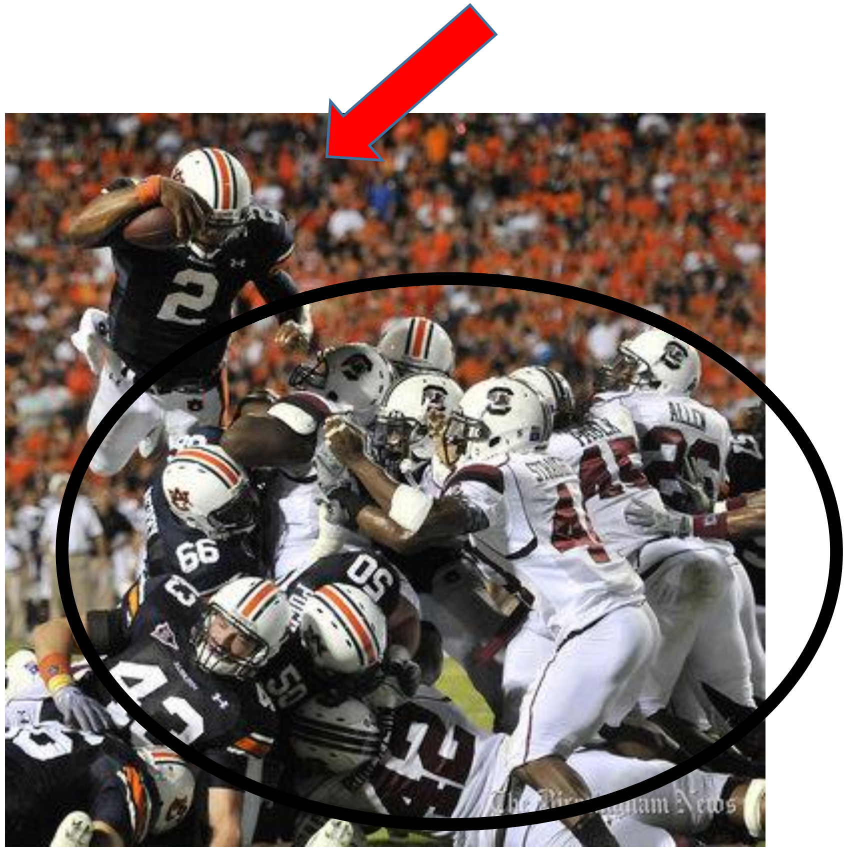
CH		SWB	
ASR	**Human**	**ASR**	**Human**
15: a	10: i	19: i	12: i
15: is	9: and	9: and	11: and
11: i	8: a	7: of	9: you
11: the	8: that	6: do	8: is
11: you	8: the	6: is	6: they
9: it	7: have	5: but	5: do
7: oh	5: you	5: yeah	5: have
6: and	4: are	4: air	5: it

Both people and machines insert “I” and “and” a lot.

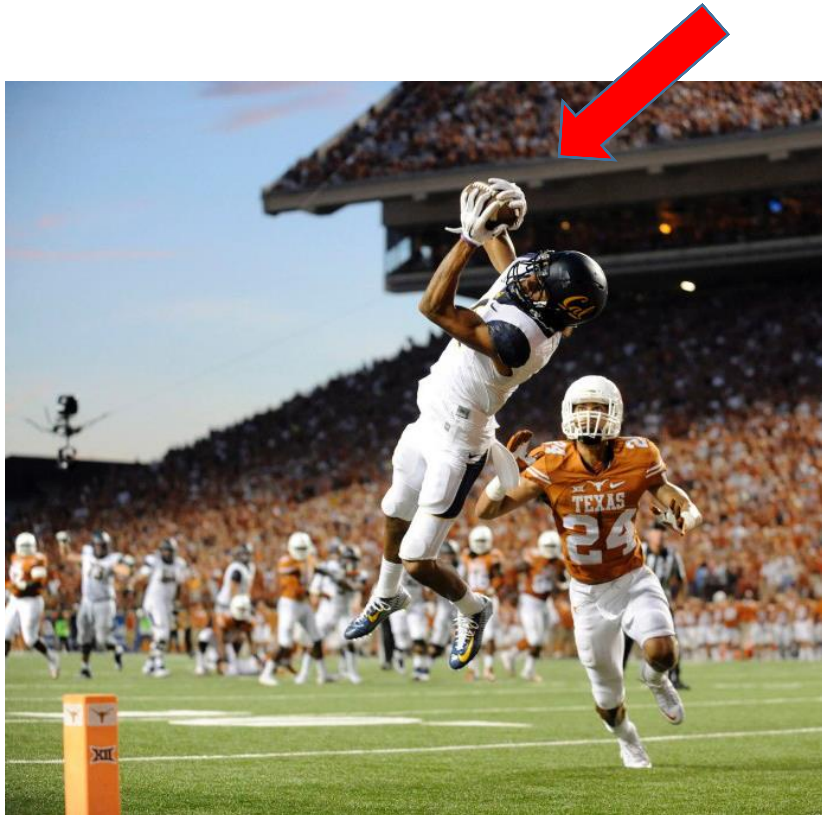
Outline

- Acoustic Modeling
- Language Modeling
- Decoding, Rescoring & System Combination
- Measuring Human Performance
- Results
- Counterpoint – Letter based CTC
- Conclusions

Two Ways to move up field



The Dive



The Pass

The CTC Alternative

- Wouldn't it be nice if we could just
  - look at the frame-level labels,
  - de-dup,
  - and read-off the transcription?
- For example, with a character model,

S – U U – P P P E E - - R G G - - O O – D D

Super Good

- CTC [Graves et al. 2006] can train a model so you can do this!

Objective Function and Gradient:

$$\text{Obj. function}: \Lambda = \sum_{\pi \in \text{alignments}} P(q \mid \pi) = \sum_{\pi \in \text{alignments}} \prod_t p^t_{q(\pi(t))}$$

$$\mathrm{P}^{\mathrm{t}}_{\mathrm{q}(\pi(\mathrm{t}))}: \text{neural net output at time } t \text{ for symbol } q(\pi(t))$$

$$\text{Gradient before softmax}: \frac{\partial \Lambda}{\partial a^t_q} = \gamma^t_q - p^t_q$$

Make the neural-net outputs look like the transcript-constrained αβ posteriors

Defining the Symbols

- Characters:
  - Generalize to new words
  - No problem with infrequent words
- Couple of issues:
  - Double-letters (e.g. “he**ll**o”) don’t work with de-duping
  - Where to insert spaces to form words (e.g. “**darkroom**” vs “**dark room**”)
- Solution:
  - introduce double-letter units (ll, oo, etc.)
  - Introduce word-initial letters (capital letters)

Explicit space character aligns acoustics to nothing.

CUDNN RNN Implementation

- Process full minibatch per CUDNN call
- 32 utterances per minibatch
- αβ computation on CPU
  - 8-way parallelization / OMP
- Best Configuration:
  - 9 Relu-RNN layers
  - Bidirectional
  - 1024 wide
  - **0.0058 xRT (!)**

```
cudnnStatus_t
cudnnRNNForwardTraining( cudnnHandle_t handle,
                         const cudnnRNNDescriptor_t rnnDesc,
                         const int seqLength,
                         const cudnnTensorDescriptor_t *xDesc,
                         const void * x,
                         const cudnnTensorDescriptor_t hxDesc,
                         const void * hx,
                         const cudnnTensorDescriptor_t cxDesc,
                         const void * cx,
                         const cudnnFilterDescriptor_t wDesc,
                         const void * w,
                         const cudnnTensorDescriptor_t *yDesc,
                         void * y,
                         const cudnnTensorDescriptor_t hyDesc,
                         void * hy,
                         const cudnnTensorDescriptor_t cyDesc,
                         void * cy,
                         void * workspace,
                         size_t workSpaceSizeInBytes,
                         void * reserveSpace,
                         size_t reserveSpaceSizeInBytes)
```

```
cudnnStatus_t
cudnnRNNBackwardData( cudnnHandle_t handle,
                      const cudnnRNNDescriptor_t rnnDesc,
                      const int seqLength,
                      const cudnnTensorDescriptor_t * yDesc,
                      const void * y,
                      const cudnnTensorDescriptor_t * dyDesc,
                      const void * dy,
                      const cudnnTensorDescriptor_t dhyDesc,
                      const void * dhy,
                      const cudnnTensorDescriptor_t dcyDesc,
                      const void * dcy,
                      const cudnnFilterDescriptor_t wDesc,
                      const void * w,
                      const cudnnTensorDescriptor_t hxDesc,
                      const void * hx,
                      const cudnnTensorDescriptor_t cxDesc,
                      const void * cx,
                      const cudnnTensorDescriptor_t * dxDesc,
                      void * dx,
                      const cudnnTensorDescriptor_t dhxDesc,
                      void * dhx,
                      const cudnnTensorDescriptor_t dcxDesc,
                      void * dcx,
                      void * workspace,
                      size_t workSpaceSizeInBytes,
                      const void * reserveSpace,
                      size_t reserveSpaceSizeInBytes )
```

Results: 2000 Hour Training

Lexicon	LM	CH	SW
N	N	26.4	17.2
N	Char NG	21.8	13.8
Y	Word NG	19.3	12.6
Y	Word NG	18.7	11.3
Y	Word RNN	**17.7**	**10.2**

Best previous result – ensemble from Hannun et al. 2014

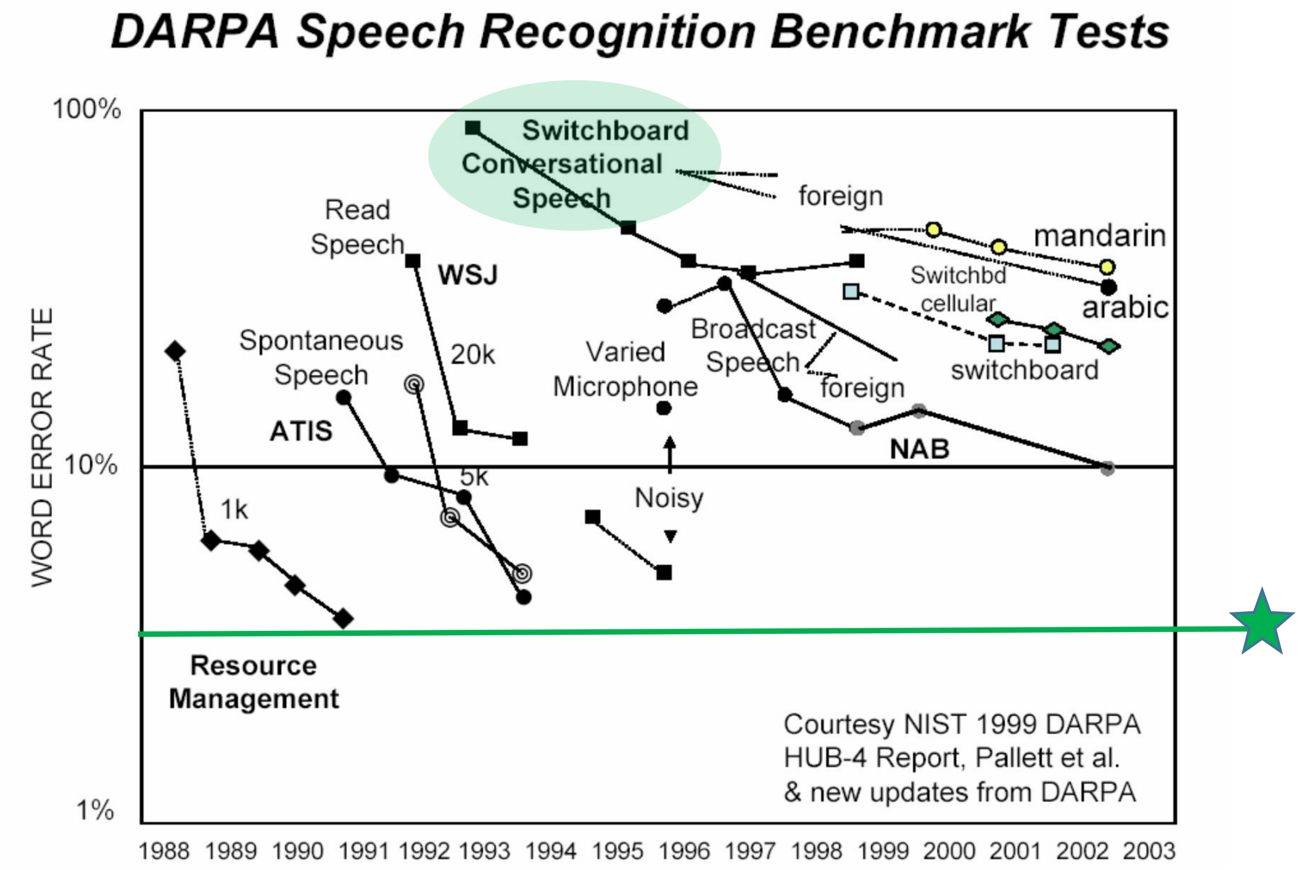
New record for CTC on Switchboard

Conclusion: Much simpler systems can produce good performance [Zweig et al., 2016]

Outline

- Acoustic Modeling
- Language Modeling
- Decoding, Rescoring & System Combination
- Measuring Human Performance
- Results
- Counterpoint – Letter based CTC
- Conclusions

Summary: Human Parity after Twenty Years



Concluding Remarks

- **Parity – Are we done?**
  - Cocktail party problem
  - Farfield
  - Robustness

Cocktail Party Problem

Concluding Remarks

- **Parity – Are we done?**
  - Cocktail party problem
  - Farfield
  - Robustness
- **What is interesting?**
  - New network structures
  - Process Simplification e.g. CTC

Concluding Remarks

- **Parity – Are we done?**
  - Cocktail party problem
  - Farfield
  - Robustness
- **What is interesting?**
  - New network structures
  - Process Simplification e.g. CTC
- **Are we stuck?**
  - CNNs! LSTMs! Attention & More.
  - The future is bright

Thank You!