Humans versus Machines: The Case of Conversational Speech Recognition

Andreas Stolcke
Speech & Dialog Research Group
Microsoft AI & Research

anstolck@microsoft.com
INTERSPEECH 2018

Speech research for emerging markets in multilingual societies

General Chair
B. Yegnanarayana

Local Organizing Chairs
Suryakanth V Gangashetty
V. Kamakshi Prasad
C. Krishna Mohan

General Co-Chairs
C. Chandra Sekhar
Shrikanth Narayan
S. Umesh
S. R. M. Prasanna

Technical Chairs
Prasanta Kumar Ghosh
Hema A Murthy
Preeti Rao
Paavo Alku

2nd - 6th September 2018
@ HICC Hyderabad
Hyderabad International Convention Centre
Acknowledgments

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

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,” and the authors who used these numbers do not describe their methodology in detail, so it is difficult to determine whether the reported error rate is accurate.

A great team effort!
Roadmap

• History of conversational speech transcription
• The Human Parity experiment
• What is human performance?
• Recognition system
• Human vs. machine error comparison
• Conclusions
The Human Parity Experiment

• Conversational telephone speech has been a benchmark in the research community for 20 years
• Can we achieve human-level performance on conversational speech?
• Top-level tasks:
  • Measure human performance
  • Build the best possible recognition system
• Analyze results
  • Inform future research
  • Pick the next challenge ...
The History
A Community Effort

Building on accumulated knowledge of many institutions!
30 Years of Speech Recognition Benchmarks

For many years, DARPA drove the field by defining public benchmark tasks.

**Read and planned speech:**
- RM
- ATIS
- WSJ

**Conversational Telephone Speech (CTS):**
- Switchboard (strangers, on-topic)
- Call Home (friends & family, unconstrained)
Prior Work

• DARPA funding ended in 2004 – a collection of papers was published in IEEE Transactions on Speech Audio and Language Processing
  • Best error rate ≈ 15% Switchboard, ≈ 40% for CallHome

• With the advent of DNNs, significant process on CTS was reported [Seide et al. 2011]

• More recent papers by IBM group, bringing WER to 6.6%, as of late 2016 [Saon et al., Interspeech]
  • IBM also quoted a 4% human error rate from the literature
Measuring Human Performance
An Early Estimate (1997)

• The 4% rumor

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).


A. Martin, 1996. Personal communication.

NIST Study of Transcriber Disagreement (2010)

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

[Glenn et al., LREC 2010]

Significant variability.

Note the bulk of the CTS training data was “quick transcribed.”
Our Human Experiment (2015)

• Skype Translator has a weekly transcription contract
  • For quality control, training, etc.

• Initial transcription followed by a second checking pass
  • Two transcribers on each speech excerpt

• One week, we added **NIST 2000 CTS evaluation** data to the pipeline
  • Speech was pre-segmented as in NIST evaluation
The Results

- Applied NIST scoring protocol
- Text normalized to minimize WER (on test set!)
- Switchboard: **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**
  - Machine error about 2x higher
  - High ASR error not just because of mismatched conditions
History of Human SWB Error Estimates

- Lippman (1997): 4%
  - based on “personal communication” with NIST, no experimental data cited
- LDC LREC paper (2010): 4.1-4.5%
  - Measured on a different dataset (but similar to our NIST eval set, SWB portion)
- Microsoft (2016): 5.9%
  - Transcribers were blind to experiment
  - 2-pass transcription, isolated utterances (no “transcriber adaptation”)
- IBM (2017): 5.1%
  - Using multiple independent transcriptions, picked best transcriber
  - Vendor was involved in experiment and aware of NIST transcription conventions
Recognition System

- Acoustic modeling
- Language modeling
- System combination
Recognition System: Highlights

• New state of the art in conversational telephone speech transcription accuracy using
• Multiple acoustic model architectures:
  • ResNet, VGG and LACE convolutional nets (CNNs)
  • Bidirectional LSTM nets
  • Speaker-adaptive modeling using i-vectors
  • Lattice-free sequence training
• Forward/backward LSTM-LM rescoring using multiple input representations
• Search for complementary acoustic model
• Confusion-network-based, weighted combination
• System achieves accuracy slightly better than human transcribers: 5.8% WER on Switchboard and 11.0% on CallHome
State of the Art has a Long History

• The current favorites: CNNs, LSTMs

• But building on key 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
Hybrid HMM/DNN

Record performance in 2011 [Seide et al.]

Hybrid HMM/NN approach still standard
But DNN model now obsolete (!)
• Poor spatial/temporal invariance

[Yu et al., 2010; Dahl et al., 2011]
Acoustic Modeling: VGG CNN

Adapted for speech from image processing [Saon et al., 2016]

Robust to temporal and frequency shifts

[Simonyan & Zisserman, 2014; Frossard 2016, Saon et al., 2016, Krizhevsky et al., 2012]
Acoustic Modeling: ResNet CNNs

Adds 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

<table>
<thead>
<tr>
<th></th>
<th>CallHome</th>
<th>Switchboard</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN</td>
<td>21.9%</td>
<td>13.4%</td>
</tr>
<tr>
<td>ResNet</td>
<td>17.3%</td>
<td>11.1%</td>
</tr>
</tbody>
</table>

1st pass decoding

[He et al., 2015]
Acoustic Modeling: LACE CNN

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

<table>
<thead>
<tr>
<th></th>
<th>CallHome</th>
<th>Switchboard</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN</td>
<td>21.9%</td>
<td>13.4%</td>
</tr>
<tr>
<td>ResNet</td>
<td>17.3%</td>
<td>11.1%</td>
</tr>
<tr>
<td>LACE</td>
<td>16.9%</td>
<td>10.4%</td>
</tr>
</tbody>
</table>

1st pass decoding

[Yu et al., 2016]
## CNN Comparison

<table>
<thead>
<tr>
<th></th>
<th>VGG Net (85M Parameters)</th>
<th>Residual-Net (38M Parameters)</th>
<th>LACE (65M Parameters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight layers</td>
<td>14</td>
<td>49</td>
<td>22</td>
</tr>
<tr>
<td>Input size</td>
<td>40x41 input</td>
<td>40x41 input</td>
<td>40x61 input</td>
</tr>
<tr>
<td>Convolutions</td>
<td>3 – conv 3x3, 96</td>
<td>3 – [conv 1x1, 64, conv 3x3, 64, conv 1x1, 256]</td>
<td>5 – conv 3x3, 128</td>
</tr>
<tr>
<td>Max pool</td>
<td>4 – [conv 1x1, 128, conv 3x3, 128, conv 1x1, 512]</td>
<td>5 – conv 3x3, 256</td>
<td></td>
</tr>
<tr>
<td>Convolutions</td>
<td>4 – conv 3x3, 192</td>
<td>6 – [conv 1x1, 256, conv 3x3, 256, conv 1x1, 1024]</td>
<td>5 – conv 3x3, 512</td>
</tr>
<tr>
<td>Max pool</td>
<td>3 – [conv 1x1, 512, conv 3x3, 512, conv 1x1, 2048]</td>
<td>5 – conv 3x3, 1024</td>
<td></td>
</tr>
<tr>
<td>Convolutions</td>
<td>4 – conv 3x3, 384</td>
<td>Average pool</td>
<td>1 – conv 3x4, 1</td>
</tr>
<tr>
<td>Max pool</td>
<td>Softmax (9000)</td>
<td>Softmax (9000)</td>
<td>Softmax (9000)</td>
</tr>
<tr>
<td>FC</td>
<td>2 – FC – 4096</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Softmax (9000)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Very deep**
- **Many parameters**
- **Small convolution patterns**
- **Processing ~ ½ second per window**
Acoustic Modeling: Bidirectional LSTMs

Stable form of recurrent neural net
Robust to temporal shifts

2nd best single model
[Hochreiter & Schmidhuber, 1997,
Graves & Schmidhuber, 2005; Sak et al., 2014]

[Graves & Jaitly ‘14]
Runtimes

<table>
<thead>
<tr>
<th></th>
<th>DNN</th>
<th>BLSTM</th>
<th>ResNet</th>
<th>LACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM Training, GPU</td>
<td>0.012</td>
<td>0.022</td>
<td>0.60</td>
<td>0.23</td>
</tr>
<tr>
<td>AM eval, GPU</td>
<td>0.0064</td>
<td>0.0081</td>
<td>0.15</td>
<td>0.081</td>
</tr>
<tr>
<td>AM eval, CPU</td>
<td>0.052</td>
<td>NA</td>
<td>11.7</td>
<td>8.47</td>
</tr>
<tr>
<td>Decoding, GPU</td>
<td>1.04</td>
<td>1.40</td>
<td>1.19</td>
<td>1.38</td>
</tr>
</tbody>
</table>

(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

GPU 10 to 100x faster than CPU
Cognitive Toolkit (CNTK) Training

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

5-10% relative improvement for Switchboard

<table>
<thead>
<tr>
<th>Configuration</th>
<th>ResNet</th>
<th>LACE</th>
<th>BLSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CH</td>
<td>SWB</td>
<td>CH</td>
</tr>
<tr>
<td>Baseline</td>
<td>17.5</td>
<td>11.1</td>
<td>16.9</td>
</tr>
<tr>
<td>i-vector</td>
<td>16.6</td>
<td>10.0</td>
<td>16.4</td>
</tr>
</tbody>
</table>

I-vectors give a fixed-length representation of a speaker’s voice [Dehak et al. 2011; Saon et al., 2013]

- 100-dim i-vectors computed per conversation side
- CNN models: i-vectors multiplied by weight matrix, serves as additional bias prior to non-linearity
- BLSTM models: i-vectors appended to each input frame
Lattice-free Discriminative Training

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

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

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

Traditionally approximated by word sequences in lattice (DAG)

Instead LFMMI uses all possible word sequences in cyclic FSA

- Simple brute force MMI (maximum mutual information criterion)
- Avoids need to generate lattices
- Alignments are always current
- Forward-backward computation can be reduced to matrix operations, run efficiently on GPUs

[Chen et al., 2006, McDermott et al., 2014, Povey et al., 2016]
Lattice-free MMI Improvements

<table>
<thead>
<tr>
<th>Configuration</th>
<th>ResNet CH</th>
<th>ResNet SWB</th>
<th>LACE CH</th>
<th>LACE SWB</th>
<th>BLSTM CH</th>
<th>BLSTM SWB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>17.5</td>
<td>11.1</td>
<td>16.9</td>
<td>10.4</td>
<td>17.3</td>
<td>10.3</td>
</tr>
<tr>
<td>i-vector</td>
<td>16.6</td>
<td>10.0</td>
<td>16.4</td>
<td>9.3</td>
<td>17.6</td>
<td>9.9</td>
</tr>
<tr>
<td>i-vector+LFMMI</td>
<td>15.2</td>
<td>8.6</td>
<td>16.2</td>
<td>8.5</td>
<td>16.3</td>
<td>8.9</td>
</tr>
</tbody>
</table>

8-14% relative improvement on SWB

- Denominator LM predicts senones based on mixed senone/phone history
- Denominator graph has 52k states and 215k transitions
- GPU-side alpha-beta computation is 0.18xRT, exclusive of NN evaluation
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, noise-contrastive estimation (NCE) training
  • Two differently initialized models, plus Ngram LM, interpolated at the word level

• LSTM LM
  • Cognitive Toolkit (CNTK), three 1000 unit layers
  • Interpolated word and letter-trigram encoding models, plus Ngram LM
Language Modeling: Results

Other tricks that help:

• Train first on in-domain and out-of-domain (Web) data, then tune on in-domain (CTS) data only

• In rescoring, forward and backward running sentence-scores are averaged

• Words outside the NN vocabulary (which is smaller than the N-gram vocab) incur a penalty – magnitude estimated on dev data

<table>
<thead>
<tr>
<th>Language model</th>
<th>PPL</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-gram LM (baseline)</td>
<td>69.4</td>
<td>8.6</td>
</tr>
<tr>
<td>+ RNNLM, CTS data only</td>
<td>62.6</td>
<td>7.6</td>
</tr>
<tr>
<td>+ Web data training</td>
<td>60.9</td>
<td>7.4</td>
</tr>
<tr>
<td>+ 2nd hidden layer</td>
<td>59.0</td>
<td>7.4</td>
</tr>
<tr>
<td>+ 2-RNNLM interpolation</td>
<td>57.2</td>
<td>7.3</td>
</tr>
<tr>
<td>+ backward RNNLMs</td>
<td>-</td>
<td>6.9</td>
</tr>
<tr>
<td>+ LSTM-LM, CTS + Web data</td>
<td>51.4</td>
<td>6.9</td>
</tr>
<tr>
<td>+ 2-LSTM-LM interpolation</td>
<td>50.5</td>
<td>6.8</td>
</tr>
<tr>
<td>+ backward LSTM-LM</td>
<td>-</td>
<td>6.6</td>
</tr>
</tbody>
</table>

WER with ResNet acoustic model
Perplexities on 1997 eval refs

LSTM-LM gives 23% relative improvement over N-gram LM
System Combination

- Lattice Generation ResNet
- Lattice Generation LACE
- Lattice Generation BLSTM
- N-gram Rescoring 500-best Generation
- LSTM rescoring, Score reweighting*

*Log-linear combination of AM, LM, pronunciation, OOV penalty etc., optimized to minimize devset WER

Word hypotheses
Posterior probabilities

“the cat sat”
N-best Confusion Network Combination

Score normalization

Weighting and summation

String Alignment

System A
-8.4 a b c
-10.4 a b d
-20.1 x e c
...

System B
-103 a b d
-245 x b c
-1245 y e c
...

System C
...

Log scores

Sentence posteriors

Combined sentence posteriors

System weights

String posteriors

[Stolcke et al., 2000]
System Selection and Weighting

• Combining all systems is not optimal
• ... and would be way to slow

• **search-rover-combo**: new SRILM tool to find best subset of systems
  • Forward greedy search (always add the system that gives the largest gain)
  • Stop when no more gain can be had
  • Reestimate system weights at each step, using EM
  • Smooth weight estimates hierarchically with previous weights (shrinkage)
Two-level System Combination

- Limited training data for system selection and weighting
  - Using old eval sets, a few thousand utterances)
- Use prior knowledge that helps reduce number of free parameter
- One strategy: two-level combination
  - Search for best subset of BLSTM systems with different meta parameters (number of senones, NN smoothing method, choice of dictionary)
  - Combine those with equal weighting
  - Treat BLSTM combo as a single system in search for all-out system combination
- First-level system selection picks systems that differ along all dimensions
  - BLSTM(1) - Baseline (no smoothing, 9k senones)
  - BLSTM(2) - With spatial smoothing [Droppo, Interspeech 2017], 9 senones
  - BLSTM(3) - With spatial smoothing, 27k senones
  - BLSTM(4) - With spatial smoothing, 27k senones, alternate dictionary
Data

• AM training: 2000h (Fisher, Switchboard, but not CallHome)
  • One system uses 300h (Switchboard only), for diversity
• LM training: Fisher, Switchboard, CallHome, UW Web data, Broadcast News
• Dev-testing, combination tuning: NIST 2002 Switchboard-1 eval set
• Evaluation: NIST 2000 (Switchboard and CallHome portions)
## Overall System Results

<table>
<thead>
<tr>
<th>System</th>
<th>N-gram LM</th>
<th>RNN-LM</th>
<th>LSTM-LM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CH</td>
<td>SWB</td>
<td>CH</td>
</tr>
<tr>
<td>ResNet, 300h training</td>
<td>19.2</td>
<td>10.0</td>
<td>17.7</td>
</tr>
<tr>
<td>ResNet</td>
<td>14.8</td>
<td>8.6</td>
<td>13.2</td>
</tr>
<tr>
<td>ResNet, GMM alignments</td>
<td>15.3</td>
<td>8.8</td>
<td>13.7</td>
</tr>
<tr>
<td>VGG</td>
<td>15.7</td>
<td>9.1</td>
<td>14.1</td>
</tr>
<tr>
<td>VGG + ResNet</td>
<td>14.5</td>
<td>8.4</td>
<td>13.0</td>
</tr>
<tr>
<td>LACE</td>
<td>15.0</td>
<td>8.4</td>
<td>13.5</td>
</tr>
<tr>
<td>BLSTM (1)</td>
<td>16.5</td>
<td>9.0</td>
<td>15.2</td>
</tr>
<tr>
<td>BLSTM (2)</td>
<td>15.4</td>
<td>8.6</td>
<td>13.7</td>
</tr>
<tr>
<td>BLSTM (3)</td>
<td>15.3</td>
<td>8.3</td>
<td>13.8</td>
</tr>
<tr>
<td>BLSTM (4)</td>
<td>14.9</td>
<td>8.3</td>
<td>13.7</td>
</tr>
<tr>
<td>BLSTM combination</td>
<td>13.2</td>
<td>7.3</td>
<td>12.1</td>
</tr>
<tr>
<td>Full system combination</td>
<td>13.0</td>
<td>7.3</td>
<td>11.7</td>
</tr>
<tr>
<td>ICASSP 2017 paper</td>
<td>13.3</td>
<td>7.4</td>
<td>12.0</td>
</tr>
<tr>
<td>Human transcribers</td>
<td></td>
<td></td>
<td>11.3</td>
</tr>
</tbody>
</table>

- LSTM-LM gives 15-20% gain over N-gram LM
- BLSTM combination alone is almost as good as the best system!
- System combination 12% relative gain over best single subsystem
- Overall, we edge just past measured human error on the same dataset

Senone-level acoustic model combination (not used in combined system)
Human/Machine: Analysis
How do human and machine transcripts differ?

• Transcripts are very close quantitatively, by overall WER

• Research questions:
  • What makes transcription easy or hard for human vs. machine?
  • Does the machine make errors that are *qualitatively* different from humans?
  • Can humans tell the difference?
Error Correlation by Speaker

Each data point is a conversation side, $N = 40$

SWB Machine WER vs. Human WER (corr: 0.65157)

$\rho = 0.65$

CH Machine WER vs. Human WER (corr: 0.73305)

$\rho = 0.73$
Error Correlation (without outliers)

Two CallHome conversations have multiple speakers on the same side, resulting in very high WER!

\[ \rho = 0.65 \]

\[ \rho = 0.80 \]
Seen vs. Unseen Switchboard Speakers

• It has been suggested that the 2000 Switchboard test set is so “easy” because most of the speakers also occur in the training set (a NIST blunder!)

• The filled dots are the *unseen* speakers

• This doesn’t seem to be the case:
  • Machine WER on unseen speakers is within the normal range
  • For the most part (3 of 4), machine WER predicts the human WER
Qualitative differences: Top Error Types

Substitutions ($\approx$ 21k words in each test set)

<table>
<thead>
<tr>
<th>CH</th>
<th>Human</th>
<th>SWB</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR</td>
<td>Human</td>
<td>ASR</td>
<td>Human</td>
</tr>
<tr>
<td>45: (%hesitation) / %back</td>
<td>12: a / the</td>
<td>29: (%hesitation) / %back</td>
<td>12: (%hesitation) / hmm</td>
</tr>
<tr>
<td>12: was / is</td>
<td>10: (%hesitation) / a</td>
<td>9: (%hesitation) / oh</td>
<td>10: (%hesitation) / oh</td>
</tr>
<tr>
<td>9: (%hesitation) / a</td>
<td>10: was / is</td>
<td>9: was / is</td>
<td>9: was / is</td>
</tr>
<tr>
<td>8: (%hesitation) / oh</td>
<td>7: (%hesitation) / hmm</td>
<td>8: and / in</td>
<td>8: (%hesitation) / a</td>
</tr>
<tr>
<td>8: a / the</td>
<td>7: bentsy / bensi</td>
<td>6: (%hesitation) / i</td>
<td>5: in / and</td>
</tr>
<tr>
<td>7: and / in</td>
<td>7: is / was</td>
<td>6: in / and</td>
<td>4: (%hesitation) / %back</td>
</tr>
<tr>
<td>7: it / that</td>
<td>6: could / can</td>
<td>5: (%hesitation) / a</td>
<td>4: and / in</td>
</tr>
<tr>
<td>6: in / and</td>
<td>6: well / oh</td>
<td>5: (%hesitation) / yeah</td>
<td>4: is / was</td>
</tr>
</tbody>
</table>

**Overall similar patterns:** short function words get confused

**One outlier:** machine falsely recognizes backchannel “uh-huh” for filled pause “uh”
- These words are acoustically confusable, have opposite pragmatic functions in conversation
- Humans can disambiguate by prosody and context
### Top Insertion and Deletion Errors

Both humans and machines insert “I” and “and” a lot. Short function words dominate the list for both.

#### Deletions

<table>
<thead>
<tr>
<th>CH</th>
<th>ASR</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>44: i</td>
<td>73: i</td>
<td></td>
</tr>
<tr>
<td>33: it</td>
<td>59: and</td>
<td></td>
</tr>
<tr>
<td>29: a</td>
<td>48: it</td>
<td></td>
</tr>
<tr>
<td>29: and</td>
<td>47: is</td>
<td></td>
</tr>
<tr>
<td>25: is</td>
<td>45: the</td>
<td></td>
</tr>
<tr>
<td>19: he</td>
<td>41: %bcack</td>
<td></td>
</tr>
<tr>
<td>18: are</td>
<td>37: a</td>
<td></td>
</tr>
<tr>
<td>17: oh</td>
<td>33: you</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SWB</th>
<th>ASR</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>31: it</td>
<td>34: i</td>
<td></td>
</tr>
<tr>
<td>26: i</td>
<td>30: and</td>
<td></td>
</tr>
<tr>
<td>19: a</td>
<td>29: it</td>
<td></td>
</tr>
<tr>
<td>17: that</td>
<td>22: a</td>
<td></td>
</tr>
<tr>
<td>15: you</td>
<td>22: that</td>
<td></td>
</tr>
<tr>
<td>13: and</td>
<td>22: you</td>
<td></td>
</tr>
<tr>
<td>12: have</td>
<td>17: the</td>
<td></td>
</tr>
</tbody>
</table>

#### Insertions

<table>
<thead>
<tr>
<th>CH</th>
<th>ASR</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>15: a</td>
<td>10: i</td>
<td></td>
</tr>
<tr>
<td>15: is</td>
<td>9: and</td>
<td></td>
</tr>
<tr>
<td>11: i</td>
<td>8: a</td>
<td></td>
</tr>
<tr>
<td>11: the</td>
<td>8: that</td>
<td></td>
</tr>
<tr>
<td>11: you</td>
<td>8: the</td>
<td></td>
</tr>
<tr>
<td>9: it</td>
<td>7: have</td>
<td></td>
</tr>
<tr>
<td>7: oh</td>
<td>5: you</td>
<td></td>
</tr>
<tr>
<td>6: and</td>
<td>4: are</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SWB</th>
<th>ASR</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>19: i</td>
<td>12: i</td>
<td></td>
</tr>
<tr>
<td>9: and</td>
<td>11: and</td>
<td></td>
</tr>
<tr>
<td>7: of</td>
<td>9: you</td>
<td></td>
</tr>
<tr>
<td>6: do</td>
<td>8: is</td>
<td></td>
</tr>
<tr>
<td>6: is</td>
<td>6: they</td>
<td></td>
</tr>
<tr>
<td>5: but</td>
<td>5: do</td>
<td></td>
</tr>
<tr>
<td>5: yeah</td>
<td>5: have</td>
<td></td>
</tr>
<tr>
<td>5: have</td>
<td>5: it</td>
<td></td>
</tr>
</tbody>
</table>
“Spot the Bot”

• Can people tell which transcripts are by machine?
• We ran an informal experiment at the last ICASSP conference
• Inspired by Turing test
Experiment: Informal results

• Subjects guessed correctly 188 / 353 times (53% accuracy)
• Not different from chance ($p \approx 0.12$, one-tailed)
• Obviously, this was not a rigorous experiment ... 
• ... but it gave us a first-hand idea of how difficult it is to tell human from machine transcription
Wrap-up
We’ve come a long way

DARPA Speech Recognition Benchmark Tests

5.8% ≈ Human performance

5k

1k

Noisy

NAB

Switchboard

mandarin

arabic

WSJ

Spontaneous Speech

Read Speech

ATIS

Broadcast Speech

Varied Microphone

Switchboard

Resource Management


1%

10%

100%

© 2016 NIST 1999 DARPA
HUB-4 Report, Pallett et al.
& new updates from DARPA

July 3, 2017
Afeka Conference for Speech Processing
Conclusions

• Human transcription performance is around 5-6%, but also varies greatly with the function of the amount of effort!
  • Multiple independent transcription passes with reconciliation would lower this further, as done by NIST for their reference transcriptions

• State-of-the-art ASR technology based on neural net acoustic and language models has reached commercial-level accuracy

• Humans and machine transcription performance is highly correlated
  • “Hard” versus “easy” speakers
  • Word types involved in most frequent errors
  • Humans are better at recognizing pragmatically relevant words (“uh” vs. “uh-huh”)
Where to go from here

• Pick harder tasks!
• Current focus (again!) = Meeting speech
  • Multiple speakers
  • Overlapping speech
  • Distant microphone capture (background noise, reverberation)
Thank You!

More Technical Details
BLSTM Spatial Regularization

Regularize with L2 norm of Hi-frequency residual

```
2-D Unrolling - Smoothed 2D = Hi-Freq
```

<table>
<thead>
<tr>
<th>Senones</th>
<th>CallHome WER (%)</th>
<th>SWB WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Baseline</td>
<td>Smoothing</td>
</tr>
<tr>
<td>9000</td>
<td>21.4</td>
<td>19.2</td>
</tr>
<tr>
<td>27000</td>
<td>20.5</td>
<td>19.5</td>
</tr>
</tbody>
</table>

5-10% relative improvement for BLSTM

[Droppo, Interspeech 2017]
MMI Denominator GPU computation

• Represent FSA of all possible state sequences as a sparse transition matrix A

• Implement exact alpha beta computations

\[
\alpha_t = (A \alpha_{t-1}) \cdot o_t \\
\beta_t = A^T \left( \beta_{t+1} \cdot o_{t+1} \right)
\]

• Execute in straight “for” loops on GPU with 
  \texttt{cusparseDcsrmv} and \texttt{cublasDdgmm}

• Beautifully simple
LM Training Trick: Self-stabilization

- Learn an overall scaling function for each layer

\[ y = Wx \quad \text{becomes:} \quad y = (\beta W)x \]

Applied to the LSTM networks, between layers.

[2016] Gahremani & Droppo
## Language Model Perplexities

<table>
<thead>
<tr>
<th>Language model</th>
<th>PPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ngram: 4gram baseline (145M ngrams)</td>
<td>75.5</td>
</tr>
<tr>
<td>RNN: 2 layers + word input</td>
<td>59.8</td>
</tr>
<tr>
<td>LSTM: word input in forward direction</td>
<td>54.4</td>
</tr>
<tr>
<td>LSTM: word input in backward direction</td>
<td>53.4</td>
</tr>
<tr>
<td>LSTM: letter trigram input in forward direction</td>
<td>52.1</td>
</tr>
<tr>
<td>LSTM: letter trigram input in backward direction</td>
<td>52.0</td>
</tr>
</tbody>
</table>

Perplexities on the 1997 eval set

- LSTM beats RNN
- Letter trigram input slightly better than word input
- Note both forward and backward running models