# uRobo:
# The Voice that (Could) Sound Like You

# uRobo: Speech Synthesis with Limited Data

- Speaker specific speech synthesis is an important area across domains
  - Medicine
  - Media Production
  - Chat bots
- How to produce speech that sounds like one person when you have limited audio of that person’s voice?

# Urobo: Speech Synthesis (ctd.)

- Multiple different speech models exist
  - Parametric
  - Neural Network End-to-End
  - Concatenative
- Concatenative
  - Uses Actual Human Speech vs. Parametric Vocoder artifacting
  - Does not require massive resources or training time

# A 4-Layer System

1) ASR/Alignment
   - Uses Kaldi to train alignment models on LibriSpeech data (collected and processed by Kaldi)
   - Final Forced Alignments used in Preprocessing Step
2) Preprocessing
   - Break each utterance into n-phones (triphones with diphone and monophone backoff)
   - Extract Features from each nphone
3) Target Feature Prediction
   - Train a Neural Model predict the features extracted in step 2
4) Concatenative Synthesis
   - Build a data set of audio units from preprocessed data
   - Select units by minimizing target and concatenation cost

# ASR

- Kaldi downloads the entire LibriSpeech corpus[2]
- Trains multiple acoustic models over 1000 hours of data
- uRobo performs forced alignment on the initial 100 clean hours of training data for future preprocessing

# Preprocessing

- Take Language Models / Alignment Information extracted by Kaldi, reformat for easier Python ingestion
- Chunk phones into n-phones (triphones with diphone and monophone backoff and single phone overlap
- Only triphones and diphones composing > 1% or .1% of total corpus respectively are chunked

# Preprocessing

Hello my name...

HH_B, EH0_I, L_I, OW1_E, M_B, AY1_E, N_B, EY1_I, M_E...

HH_B, EH0_I, L_I, [OW1_E, M_B],

[M_B, AY1_E],

[AY1_E, N_B],

[N_B, EY1_I],

[EY1_I, M_E],

M_E...

# Preprocessing

Hello my name...

HH_B, EH0_I, L_I, OW1_E, M_B, AY1_E, N_B, EY1_I, M_E...

HH_B, EH0_I, L_I, [OW1_E, M_B],
[M_B, AY1_E],
[AY1_E, N_B],
[N_B, EY1_I],
[EY1_I, M_E],
M_E...

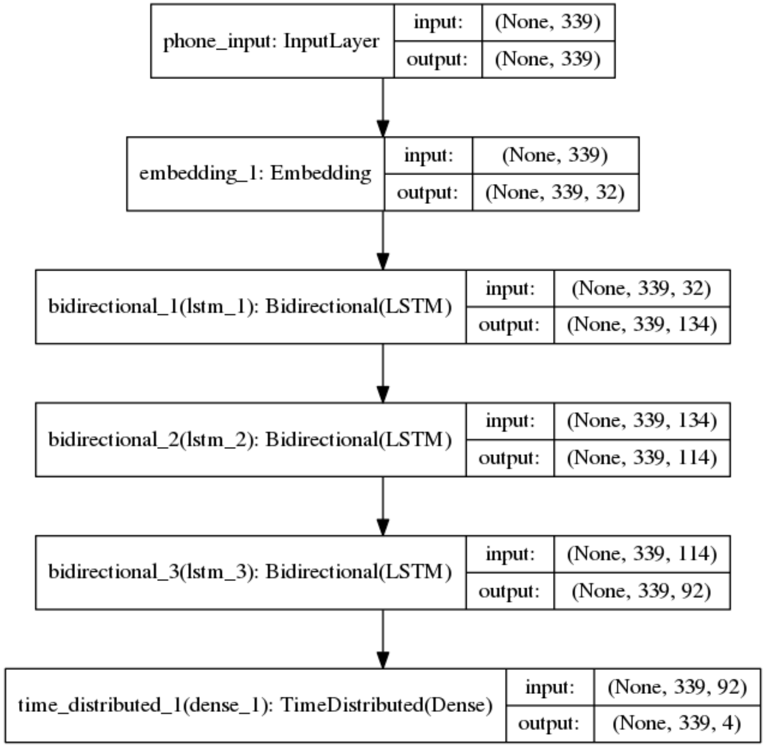
# Preprocessing

- Extract target features
  - Duration
  - Initial Phone $F_0$
  - Final Phone $F_0$
  - Energy
- Features are scaled by z-score respective to speaker stats[3]

$$\hat{feat}_{spk} = \frac{feat_{spk} - \mu_{spk}}{\sigma_{spk}}$$

# Target Feature Prediction

- N-phone indexes are fed into a 3-layer bidirectional LSTM neural network and trained on features[4]



# Concatenative Synthesis

- Classic Concatenation involves “unit selection” from a data set of audio units thought to represent a given utterance
- These are spliced together to form a final file
- Minimization of
  - target cost $C^t$ (what should the unit sound like)
  - concatenation cost $C^c$ (how well does it follow the preceding unit)[1]
- Viterbi algorithm (dynamic programming) used to determine the final sequence of units
- Logarithmic crossfade across overlapping monophones used during concatenation

# Concatenative Synthesis (ctd.)

- Target Cost was the sum of the absolute difference between Z-Scored features (duration, 1st phone $f_0$, Last phone $f_0$, energy)

$$C^t(t_i, u_i) = \sum_{j=1}^{p} |t_{i_j} - u_{i_j}|$$

- Concatenation Cost was the sum of the absolute difference between Z-Scored features (overlapping/joining phone $f_0$, energy)

$$C^c(u_{i-1}, u_i) \sum_{j=1}^{q} |u_{i-1_j} - u_{i_j}|$$

# Speech

```
What would you like me to say? hello my name is alice
Initializing viterbi matrix.
Going through 389 phone 0 candidates.
Going through 12 phone 1 candidates.
Going through 1 phone 2 candidates.
Going through 2 phone 3 candidates.
Going through 9 phone 4 candidates.
Going through 2 phone 5 candidates.
Going through 5 phone 6 candidates.
Going through 5 phone 7 candidates.
Going through 109 phone 8 candidates.
Going through 7 phone 9 candidates.
Going through 14 phone 10 candidates.
Going through 182 phone 11 candidates.
Going through 5 phone 12 candidates.
Going through 331 phone 13 candidates.
Going through 166 phone 14 candidates.
FINAL COST: 47.6197339907
FINAL PHONES: ['HH_B', 'EH0_I', ['L_I'], ['OW1_E', 'M_B'], ['M_B', 'AY1_E'], ['AY1_E', 'N_B'], ['N_B', 'EY
1_I'], ['EY1_I', 'M_E'], 'M_E', ['IH1_B', 'Z_E'], ['Z_E', 'AE1_B'], 'AE1_B', ['L_I', 'IH0_I'], 'IH0_I', 'S
_E']
```

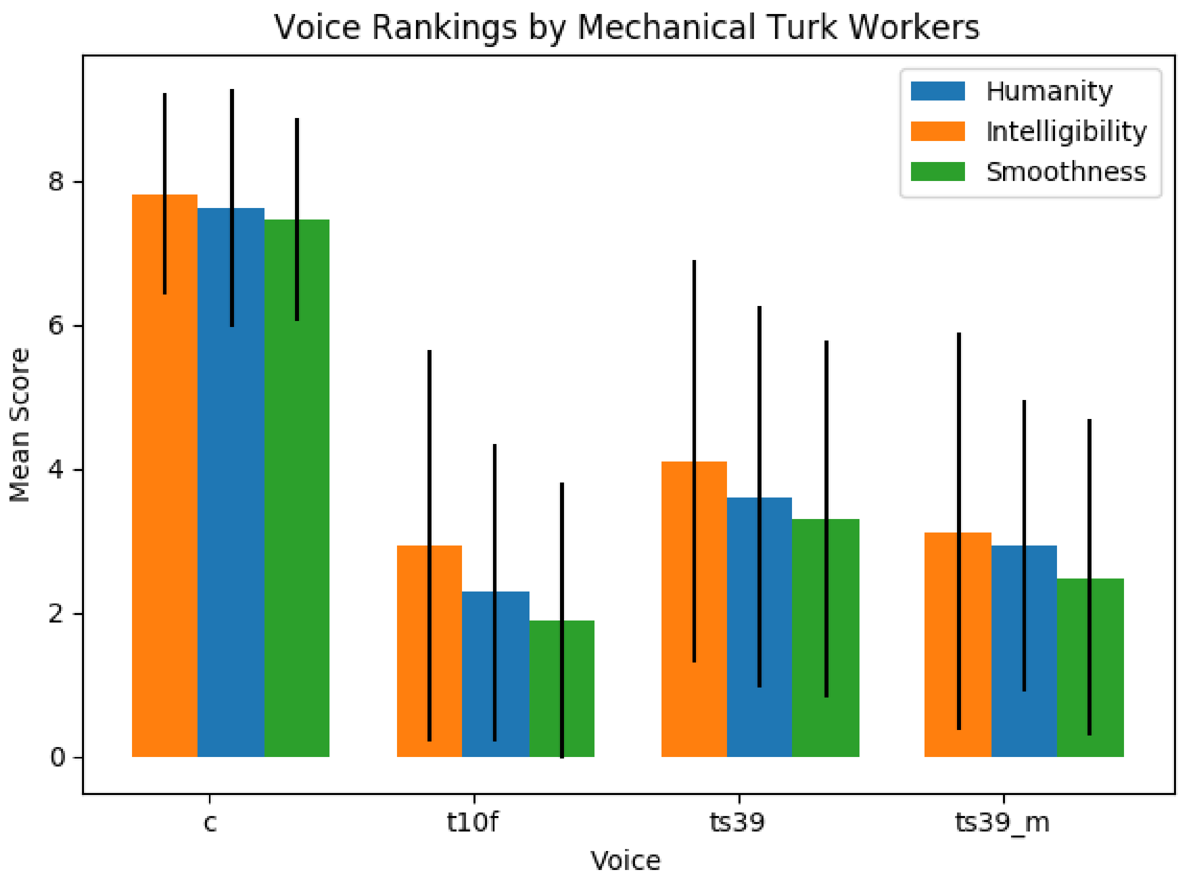
# Experiment

- To determine the effectiveness of the architecture, three voices were tested against a control voice *c*
  - *t10f*: 10 hours of audio units from different female speakers
  - *ts39*: 25 minutes of audio units from a single female speaker
  - *ts39_m*: Same as above but only utilizing monophones rather than n-phones

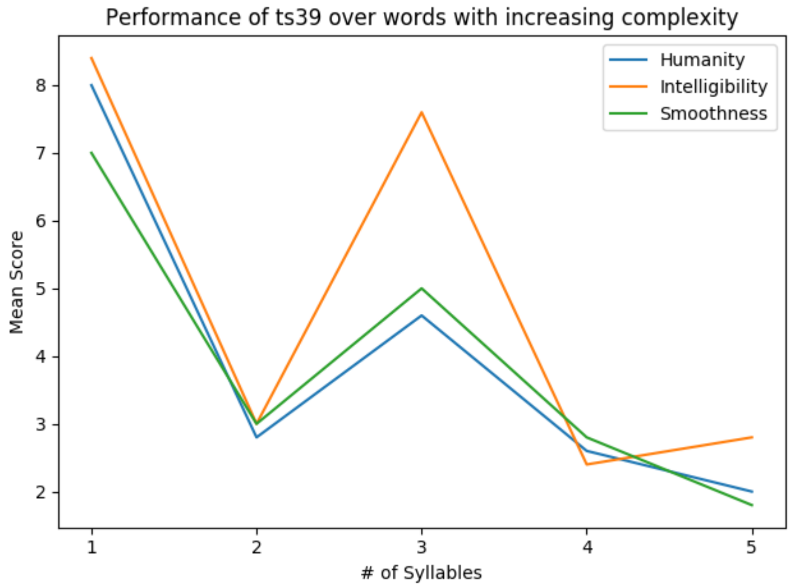
# Experiment (ctd.)

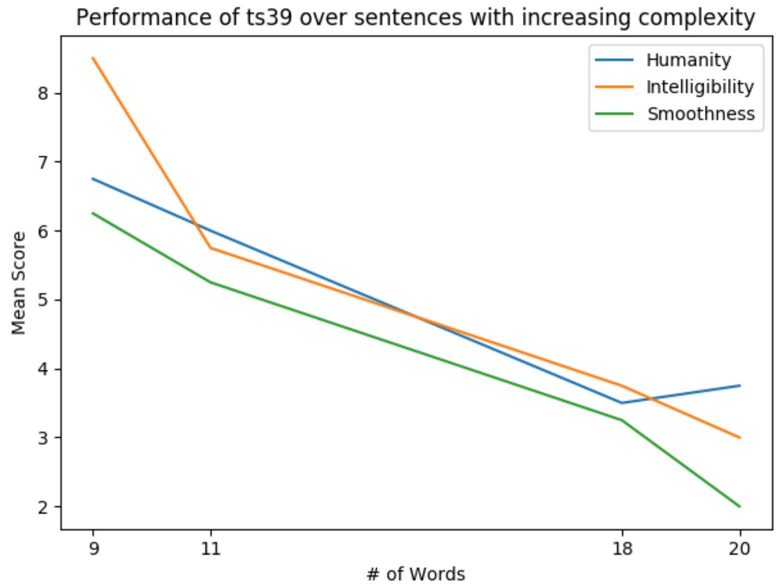
- 10 utterances generated from the 3 voices
  - 5 words of 1-5 syllables each (in ascending order)
  - 5 sentences of increasing complexity
  - All compared against a human voice from the LibriSpeech test corpus
- 5 unique workers listened to each utterance
  - Judged the utterance based on
    - Intelligibility
    - Humanity (how “human” it sounded)
    - Smoothness
- A total of 28 workers participated

# Experiment Results



# Experiment Results (ctd.)



# Conclusions/Avenues for Future Work

- Performance surprising given the low amount of base unit data used to synthesize the voice
- uRobo is a complex system and could be tuned at many different layers to show improvement
  - Different treatment of silences / sequence to sequence of words to phones
  - HMM/GMM model for preselection
- Different voice models may also help
  - Single Speaker with 10 hours of Data
    - Increased Variation of pronunciations over 25 minutes
    - Better phonetic coverage
  - Multi-Speaker with pitch contour normalization

# References

[1] A. J. Hunt and A. W. Black, “Unit selection in a concatenative speech synthesis system using a large speech database,” in *Proceedings of the Acoustics, Speech, and Signal Processing, 1996. On Conference Proceedings., 1996 IEEE International Conference- Volume 01*, ser. ICASSP ’96. Washington, DC, USA: IEEE Computer Society, 1996, pp. 373–376. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.1996.541110

[2] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The kaldi speech recognition toolkit,” in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[3] H. Beigi, *Fundamentals of Speaker Recognition*. Springer Publishing Company, Incorporated, 2011.

[4] F. Chollet et al., “Keras,” https://github.com/fchollet/keras, 2015