Voice Translator (Project Proposal)
group: heming han, ian calloway, sam tenka, seth raker
date: 2016-10-15
descr: Proposal for Eecs 351 Project

0. Description
The Voice Translator imitates human speech. More precisely, it solves an analogy problem: what is to Danny saying ``colorless green ideas'' as Sarah saying ``the quick brown fox'' is to Danny saying ``the quick brown fox''? The answer will be an audio signal that sounds like Sarah saying ``colorless green ideas''. See Figure [6.0].

1. Potential Challenges
1.0. Learning and validation may be problematic, since there's no obvious psychologically meaningful distance function on audio files. We'll have to validate the system piece-by-piece, and use heuristics for end-to-end validation. For training, we may explore recent unsupervised metric-learning techniques such as Generative Adversarial Networks [5.3].

1.1. The number and nature of voice qualities is, to us, currently unknown. It will be crucial that we review the speech-processing literature, e.g. [5.1]. Black-box methods robust to domain, such as neural nets, may overcome this problem [5.2].

1.2. Since phonemes interact, concatenation is nontrivial. As a reach goal, we'll train our model to contextualize phonemes given their neighbors, and, likewise, use context to better filter out non-phoneme noise.

2. Data
The freely available CMU Arctic dataset contains paired speech clips of ~1000 sentences, each read by the same 7 humans [5.4]. Of those 7 humans, 3 speak with standard american accents, and 4 speak with ``other accents'', providing variety suitable to training a speech-transcriber. We are currently looking for larger datasets to train the synthesis component of our system.

To augment the above with a richer variety of voices and words, we can record our friends: 150 words per minute x 60 minutes per hour x 0.25 hours per friend x 10 friends per teammate x 3 teammates = 67,500 words. Compare to 42,000 distinct recognized lemmas for 20-year-old English speakers [5.0]: ignoring multiplicity, the data is on the order of magnitude of one's vocabulary.

3. System Design

The following tools are tentative: we demonstrate that tools exist, but refrain from yet committing to any one approach.

3.0 Algorithm Ideas

We can factor the voice-translation problem into steps, with potential tools listed (see Figure [6.1]):

3.0.0. Analyze input audio into style/content components (see Figure [6.2])

→ Segmentation by analyzing pauses and piecewise-constant regression; cepstral coefficients; classification by convolutional neural network; encoding via generative adversarial networks.

The above has many alternatives. For instance, segmentation-then-classification can be replaced by transcription via Hidden Markov Models.

3.0.1. Synthesize style/content components into output audio (see Figure [6.3])

→ Each of the tools listed in [3.0] has a natural inverse.

Here, the *`style'* component will be a fixed-length vector of psychologically meaningful voice qualities such as pitch, nasality, vowels-height, and the degree to which r's are rolled. On the other hand, the *`content'* component will be a variable-length transcription of the audio into a discrete phonetic alphabet. Thus, we have two further implicit design tasks:

3.0.2. Find natural representations for style and content.

→ This subtask will profit from extensive review of literature. So far, our readings have divided into three main topics: phonetic transcription and representation, representation of raw audio, and techniques for dimensionality reduction of rich data. The representatives most familiar to us in each respective category are: the International Phonetic Alphabet [5.5], the cepstral transform [5.1], and deep autoencoder networks [5.6].

3.0.3. Pre-compute a good prior distribution on the space of voices.

→ Our system will fail to complete the analogy proposed in section [0] without an understanding of voices beyond those inputted in runtime. We expect that a significant portion of the project will lie in pre-runtime training of learning algorithms on fixed datasets. See section [2] for data discussion.

3.1 Development Tools

We'll use *Audacity* and *MATLAB* to explore audio data and the qualitative nature of speech in the time and frequency domains. We'll prototype signal processing algorithms in *MATLAB* and *Python/numpy/audioop/wave*, or, for neural networks, *Python/Keras/TensorFlow* or *Lua/Torch*. For speed, we will eventually migrate to a *C++* backend, wrapped in a Python GUI interface via *SWIG*. Our project will be hosted and version-controlled on GitHub, and proudly described on a website written in pure HTML.

4. Timeline

The project is due on 2016-12-12, which is 8 weeks from the date of this proposal's submission, 2012-10-19. Leaving a buffer of 2 weeks for exams and breaks, we plan to complete our project in 6 weeks, labeled [0, 6), pacing ourselves at two major goals per week:

	wk -2	wk -1	wk 0	wk 1	wk 2	wk 3	wk 4	wk 5
Goal A	Choose Team	Review Literature	Baseline Transcriber	Baseline Voice Representat'n	Baseline Synth	Test & improve system; Try out alternative techniques.		
Goal B	Brainstorm System	Acquire Data	Explore Data	Evaluation Pipeline	Baseline System	Optimize (C++)	Create Website	Present System

With a team of 4 motivated members, the above timeline seems ambitious but realistic.

5. References
We find the following work useful:

5.0. Exploring spoken language richness at the symbol (not audio) level:
http://journal.frontiersin.org/article/10.3389/fpsyg.2016.01116/full

5.1. Cepstral transform for characterizing voices:
http://ieeexplore.ieee.org/document/1455016/?arnumber=1455016&tag=1 2.

5.2. Style/content separation via Deep Neural Nets:
https://arxiv.org/abs/1508.06576

5.3. Rich-from-rich semi-supervised learning via GANS:
http://arxiv.org/abs/1605.05396

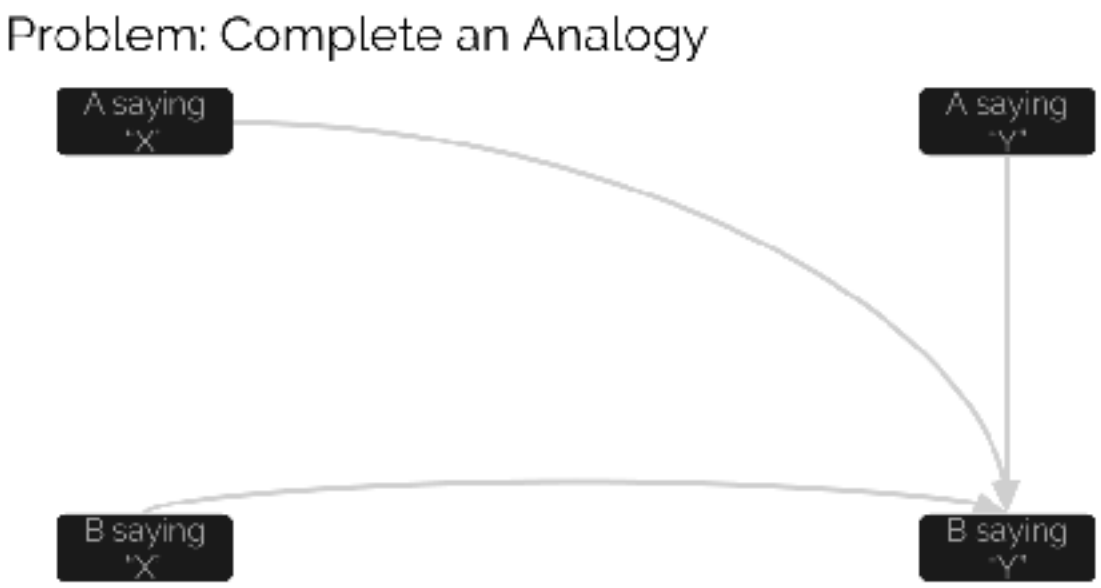
5.4. Paired Speech Data
http://www.festvox.org/cmu_arctic/cmu_arctic_report.pdf

5.5. International Phonetic Alphabet
https://www.internationalphoneticassociation.org/sites/default/files/IPA_Kiel_2015.pdf

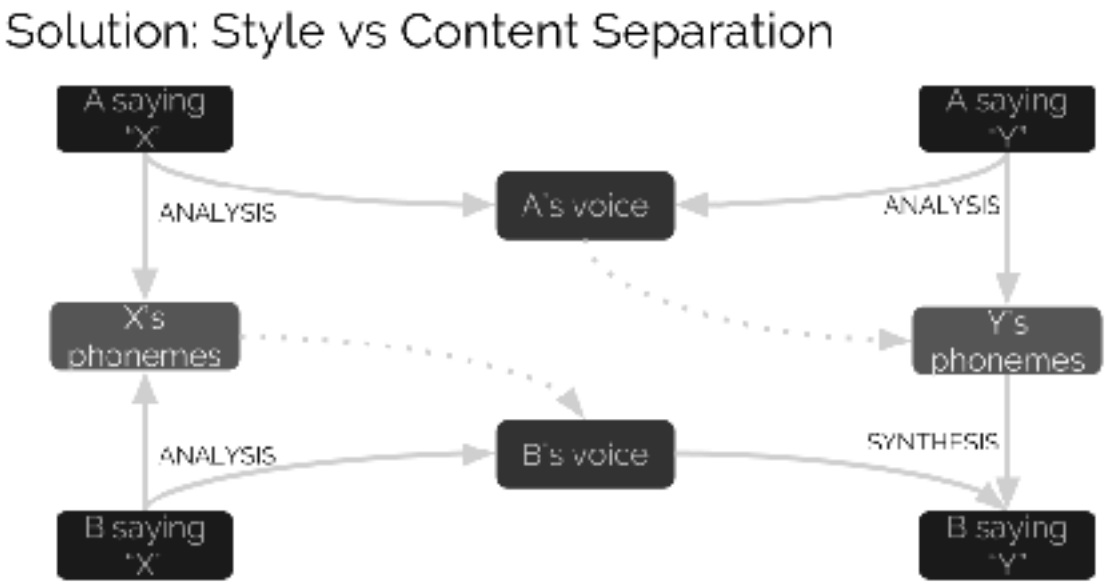
5.6. Autoencoders for Speech
5.6.0. Hybrid Hidden Markov Model and Deep Network:
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/dbn4lvcsr-transaslp.pdf
5.6.1. Extreme Speech Compression via Deep Belief Networks
http://www.cs.toronto.edu/~asamir/papers/is10_2.pdf

6. Figures

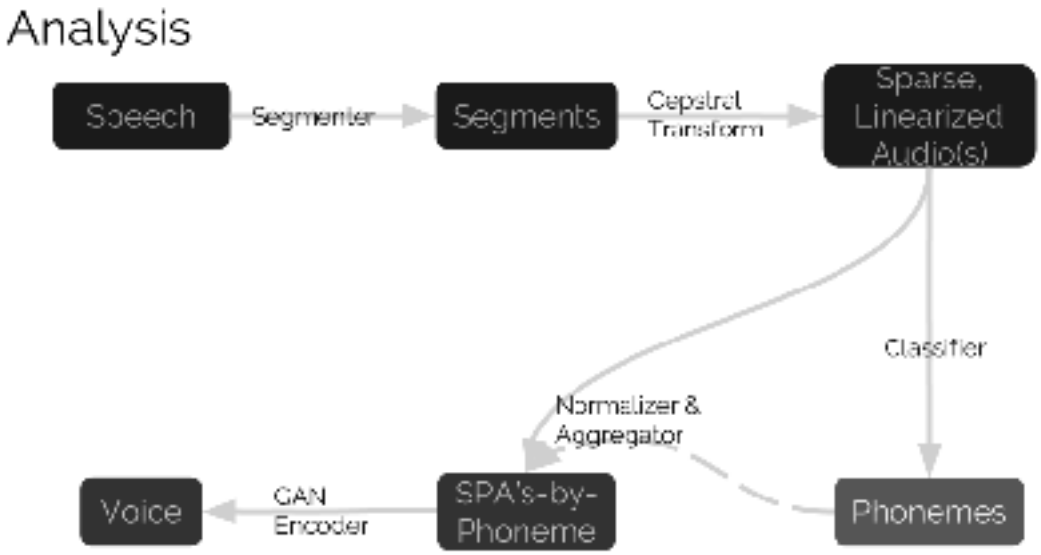
6.0



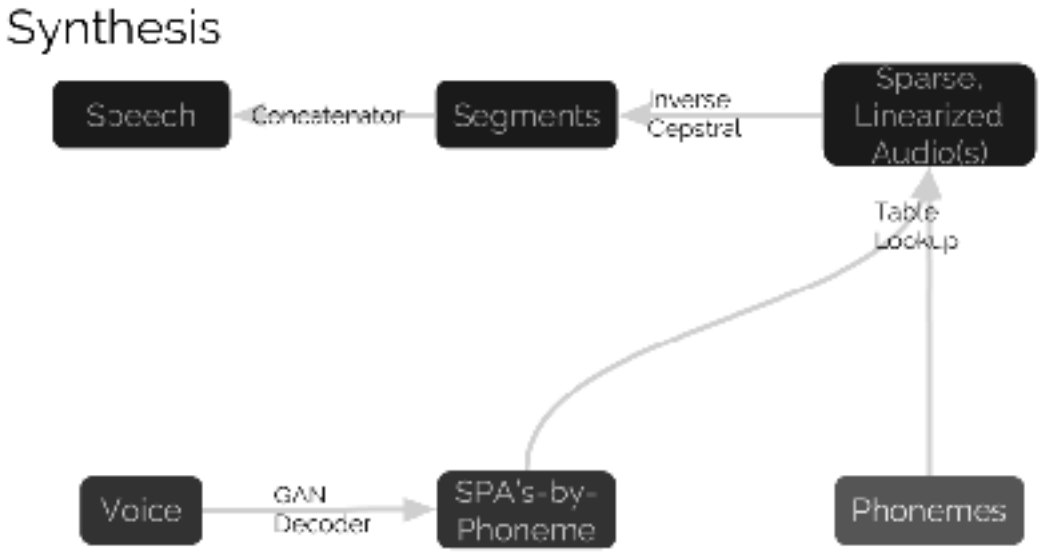
6.1



6.2


Analysis
Speech
Segmenter
Segments
Cepstral Transform
Sparse, Linearized Audio(s)
Classifier
Normalizer & Aggregator
Voice
GAN Encoder
SPA's-by-Phoneme
Phonemes

6.3


Synthesis
Speech
Concatenator
Segments
Inverse Cepstral
Sparse, Linearized Audio(s)
Table Lookup
Voice
GAN Decoder
SPA's-by-Phoneme
Phonemes