
Kallirroi Georgila

02.27.2013
Typical dialogue system architecture

User

- Speech Recognizer
- Natural Language Understanding
- Natural Language Generation
- Dialogue Manager
- Speech Synthesizer
Overview

- Statistical approaches to dialogue management
  - Reinforcement learning
  - User simulation
  - User simulation evaluation metrics
- Speech synthesis for dialogue systems
  - Unit-selection speech synthesis
  - HMM speech synthesis
  - Expressive conversational speech synthesis
  - Amazon Mechanical Turk evaluation study
Overview

- Statistical approaches to dialogue management
  - Reinforcement learning
  - User simulation
  - User simulation evaluation metrics

- Speech synthesis for dialogue systems
  - Unit-selection speech synthesis
  - HMM speech synthesis
  - Expressive conversational speech synthesis
  - Amazon Mechanical Turk evaluation study
Motivation for statistical dialogue management

- The *dialogue strategy (or dialogue policy)* of a dialogue system decides on which action the system should perform at a given state in the dialogue.

- Designing the dialogue strategy of a dialogue system requires a major development effort:
  - problems with hand-crafted approaches
    - it is hard to predict all possible scenarios
    - not reusable
Hand-crafted dialogue policy

User says: Italian food (ASR conf=c)

Get slot X1 (what type of food?)

If $c \geq 0.9$

If $0.7 \leq c < 0.9$

If $0.5 \leq c < 0.7$

No

No

No

Yes

Yes

Yes

Get slot X2 (what type of service?)

Implicit confirm slot X1 (okay Italian food)

Get slot X2 (what type of service?)

Explicit confirm slot X1 (did you say Italian food?)

User says: Italian food (ASR conf=c)
Flow chart for typical commercial dialogue system

Typical commercial spoken dialog system contains ~100 pages of flowchart

Flow chart taken from Williams, invited talk at ASRU 2009
The dialogue strategy (or dialogue policy) of a dialogue system decides on which action the system should perform at a given state in the dialogue.

Designing the dialogue strategy of a dialogue system requires a major development effort:
- problems with hand-crafted approaches
  - it is hard to predict all possible scenarios
  - not reusable
- problem with statistical approaches to dialogue management
  - require large amounts of data

The evaluation of a dialogue system is an expensive and time consuming process:
- the system has to be tested each time a change is made

We need to simplify and make more efficient the design and evaluation process of dialogue systems.
What is a simulated user?

- A *simulated user* is a conversational agent designed to simulate the behavior of a real user when interacting with a dialogue system.
- Given a dialogue context the simulated user generates an action appropriate in this context.
  - example user action: `greeting_opening, provide_price`
    - which translates to the following utterance:
      *Hello, I’d like to find a cheap restaurant.*
- Simulated users should generate a variety of actions in each dialogue context to capture the richness of real user behavior.
- Simulated users are typically trained on a small corpus of human-machine interactions.
Overall architecture of dialogue simulation

1. User Simulation
   - Utterance Simulation
   - Action Simulation

2. ASR Simulation

3. NLU
   - Dialogue Manager
   - System Action

4. Dialogue System

   Reinforcement Learning
Reinforcement learning

- *Reinforcement learning* is learning by interaction and trial and error
- The system interacts with the simulated user
- Reward function
  - The system is rewarded for good actions
    - e.g., when the result of the system action is that the user provides or confirms information
  - The system is penalized for bad actions
    - e.g., actions that lead to very long dialogues
Markov decision processes (MDPs)

- A *Markov Decision Process (MDP)* is defined as a tuple \((S, A, P, R, \gamma)\)
  - \(S\) is the set of states representing different contexts which the agent may be in
  - \(A\) is the set of actions of the agent
  - \(P : S \times A \rightarrow P(S, A)\) is the set of transition probabilities between states after taking an action
  - \(R : S \times A \rightarrow \mathbb{R}(S, A)\) is the reward function
  - \(\gamma\) is a discount factor weighting long term rewards
- The goal is to find the optimal policy \(\pi : S \rightarrow A\)
Bellman equations

\[ V^\pi(s) = \sum_a \pi(s,a) \sum_{s'} P^{a}_{ss'} [ R^{a}_{ss'} + \gamma V^\pi(s') ] \]

\[ Q^\pi(s,a) = \sum_{s'} P^{a}_{ss'} [ R^{a}_{ss'} + \gamma \sum_{a'} \pi(s,a') Q^\pi(s',a') ] \]
Bellman optimality equations

\[ V^*(s) = \max_{\pi} V^\pi(s) \]

\[ V^*(s) = \max_a \sum_{s'} P_{ss'}^a \left[ R_{ss}^a + \gamma V^*(s') \right] \]

\[ Q^*(s,a) = \max_{\pi} Q^\pi(s,a) \]

\[ Q^*(s,a) = \sum_{s'} P_{ss'}^a \left[ R_{ss}^a + \gamma \max_{a'} Q^*(s',a') \right] \]
Temporal difference learning

With temporal difference learning methods you only need to wait until the next step to update the $V$ or $Q$ values.

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$
**Q-learning algorithm**

Initialize $Q(s,a)$ arbitrarily

Repeat (for each episode):
  - Initialize $s$
  
    Repeat (for each step of episode):
      - Choose $a$ from $s$ using policy derived from $Q$ (e.g., e-greedy)
      - Take action $a$, observe $r, s'$
      - $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_a Q(s',a) - Q(s,a)]$
      - $s \leftarrow s'$;
    
    until $s$ is terminal

**Watkins, 1989**

**Sutton and Barto, 1998**

---

**e-greedy:**
Choose random action with probability $e$ (exploration)
Choose best action based on current policy with probability $1-e$ (exploitation)
Overall architecture of dialogue simulation

- User Simulation
  - Utterance Simulation
  - Action Simulation

- ASR Simulation

- NLU
  - Dialogue Manager
  - System Action

- Dialogue System

- Reinforcement Learning
N-gram based user simulation

example 3-grams:

user,provide_price system,explicit_confirm_price → user,confirm_pos_price 0.5
user,provide_price system,explicit_confirm_price → user,confirm_neg_price 0.4
user,provide_price system,explicit_confirm_price → user,[ ],[ ] 0.1

Eckert, Levin and Pieraccini, 1997
Georgila, Henderson and Lemon, 2005; 2006
Agenda based user simulation

Constraints =

| type=bar |
| drinks=beer |
| area=central |

Requests =

| name= |
| address= |
| phone= |

Sys₀ = Hello, how may I help you?

A₁ =

inform(type=bar)
inform(drinks=beer)
inform(area=central)
request(name)
request(address)
request(phone)
bye()

Sys₁ = Okay a wine bar. What price range?

A₂ =

negate(drinks=beer)
inform(pricerange=cheap)
inform(area=central)
request(name)
request(address)
request(phone)
bye()

User₁ = I’m looking for a nice bar serving beer.

User₂ = No, beer please!

Schatzmann and Young, 2009
Other types of user simulation

- Graph-based models (Scheffler and Young, 2001)
- Bayesian networks (Pietquin, 2004; Pietquin and Dutoit, 2005)
- Hidden Markov Models (HMMs) (Cuayáhuitl et al., 2005)
- Cluster-based user simulations (Rieser and Lemon, 2006)
- Knowledge consistent user simulations (Ai and Litman, 2007)
- Unsupervised learning (Lee and Eskenazi, 2012)
Overall architecture of dialogue simulation

User Simulation

Utterance Simulation

Action Simulation

ASR Simulation

NLU

Dialogue Manager

System Action

Dialogue System

Reinforcement Learning
What is ASR simulation?

- Given a source utterance an ASR simulation model generates a “scrambled” target utterance which is similar to the output of a real speech recognizer.

- Source utterance

  *I want an expensive hotel please*

- “Scrambled” target utterance

  *one inexpensive hotel please*

_Schatzmann, Thomson and Young, 2007_
ASR simulation

- Phonetic confusions using information from the acoustic model (Fossler-Lussier et al., 2002; Deng et al., 2003; Pietquin et al., 2004)
  - advantages: domain independent
  - disadvantages: computationally expensive, requires a large amount of training data

- Measure the confusability of each word in a corpus by counting how many other words it is confused with (Pietquin and Dutoit, 2005)
  - advantages: very easy to implement
  - disadvantages: does not take into account the context of each word, requires domain dependent data i.e., pairs of ASR outputs and human transcriptions

- A computationally inexpensive approach that takes into account the context of each word (Schatzmann et al., 2007; Roque et al., 2010)
  - advantages: computationally inexpensive
  - disadvantages: requires a substantial amount of domain dependent data i.e., pairs of ASR outputs and human transcriptions

- Phonetic confusions from alignments on the phoneme level using Weighted Finite-State Transducers (Stuttle et al., 2004; Feng Tan et al., 2010)
  - advantages: domain independent
  - disadvantages: computationally expensive
Overall architecture of dialogue simulation

- User Simulation
  - Utterance Simulation
  - Action Simulation
- ASR Simulation
- NLU
  - Dialogue Manager
- System Action
- Dialogue System

Reinforcement Learning
## Learning dialogue policies – Example system and user actions

<table>
<thead>
<tr>
<th>System actions</th>
<th>User actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>request_price</td>
<td>provide_price</td>
</tr>
<tr>
<td>request_area</td>
<td>provide_area</td>
</tr>
<tr>
<td>request_foodtype</td>
<td>provide_foodtype</td>
</tr>
<tr>
<td>explicit_confirm_price</td>
<td>confirm_pos_price</td>
</tr>
<tr>
<td>explicit_confirm_area</td>
<td>confirm_neg_price</td>
</tr>
<tr>
<td>explicit_confirm_foodtype</td>
<td>confirm_pos_area</td>
</tr>
<tr>
<td>implicit_confirm_price</td>
<td>confirm_neg_area</td>
</tr>
<tr>
<td>implicit_confirm_area</td>
<td>confirm_pos_foodtype</td>
</tr>
<tr>
<td>implicit_confirm_foodtype</td>
<td>confirm_neg_foodtype</td>
</tr>
<tr>
<td>hello</td>
<td>hello</td>
</tr>
<tr>
<td>bye</td>
<td>bye</td>
</tr>
<tr>
<td>release_turn</td>
<td>release_turn</td>
</tr>
<tr>
<td>null</td>
<td></td>
</tr>
</tbody>
</table>
## Learning dialogue policies – Example state representation

<table>
<thead>
<tr>
<th>State features</th>
</tr>
</thead>
<tbody>
<tr>
<td>status of slot “price” (null, filled, confirmed)</td>
</tr>
<tr>
<td>status of slot “area” (null, filled, confirmed)</td>
</tr>
<tr>
<td>status of slot “foodtype” (null, filled, confirmed)</td>
</tr>
<tr>
<td>previous user action excluding “release_turn” (null, provide_price,</td>
</tr>
<tr>
<td>provide_area, provide_foodtype, confirm_pos_price, confirm_neg_price,</td>
</tr>
<tr>
<td>confirm_pos_area, confirm_neg_area, confirm_pos_foodtype, confirm_neg_type,</td>
</tr>
<tr>
<td>hello, bye</td>
</tr>
</tbody>
</table>

324 possible states, 3888 possible state-action pairs
Learning dialogue policies – Example reward functions

<table>
<thead>
<tr>
<th></th>
<th>Example reward function 1</th>
<th>Example reward function 2</th>
<th>Example reward function 3</th>
<th>Example reward function 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Slot filled</strong></td>
<td>+100</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
</tr>
<tr>
<td><strong>Slot confirmed</strong></td>
<td>+100</td>
<td>+200</td>
<td>not used</td>
<td>not used</td>
</tr>
<tr>
<td><strong>Dialogue length</strong></td>
<td>-5 per system turn</td>
<td>-5 per system turn</td>
<td>-5 per system turn</td>
<td>-5 per system turn</td>
</tr>
<tr>
<td><strong>All slots filled</strong></td>
<td>not used</td>
<td>not used</td>
<td>+100</td>
<td>not used</td>
</tr>
<tr>
<td><strong>All slots confirmed</strong></td>
<td>not used</td>
<td>not used</td>
<td>+100</td>
<td>+200</td>
</tr>
</tbody>
</table>
A **Partially Observable Markov Decision Process (POMDP)** is defined as a tuple \((S, A, P, R, O, Z, \gamma, b_0)\)

- \(S\) is the set of states representing different contexts which the agent may be in
- \(A\) is the set of actions of the agent
- \(P: S \times A \rightarrow P(S, A)\) is the set of transition probabilities between states after taking an action
- \(R: S \times A \rightarrow \mathbb{R}(S, A)\) is the reward function
- \(O\) is the set of observations that the system can receive about the world
- \(Z: S \times A \rightarrow Z(S, A)\) is the set of observations probabilities
- \(\gamma\) is a discount factor weighting long term rewards
- Belief state \(b\): distribution over states, \(b(s_i)\) is the probability of being in state \(s_i\) with initial belief state \(b_0\)

The goal is to find the optimal policy \(\pi : S \rightarrow A\)
POMDP Dialer: call from 2123874000

<table>
<thead>
<tr>
<th>Previous system action</th>
<th>Belief State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sorry, first and last name?</td>
<td>Remaining mass [0 partition(s)]</td>
</tr>
<tr>
<td></td>
<td>jason williams florham_park nj (usa)</td>
</tr>
<tr>
<td></td>
<td>jason fong columbia, md (usa)</td>
</tr>
<tr>
<td></td>
<td>juan dong north_sydney, au (iaus)</td>
</tr>
<tr>
<td></td>
<td>jason downing sacramento, ca (usa)</td>
</tr>
<tr>
<td></td>
<td>jason kan englewood, co (usa)</td>
</tr>
<tr>
<td></td>
<td>jason hendrix houston, tx (usa)</td>
</tr>
<tr>
<td></td>
<td>zhesheng huang middletown, nj (usa)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State Features</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Best name</td>
<td>✖</td>
</tr>
<tr>
<td>Best phone type</td>
<td>✖</td>
</tr>
<tr>
<td>Phones available</td>
<td>both</td>
</tr>
<tr>
<td>Name confirmed?</td>
<td>no</td>
</tr>
<tr>
<td>Name is ambiguous?</td>
<td>no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Allowed Actions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AskName</td>
<td>Sorry, first and last name?</td>
</tr>
<tr>
<td>AskPhoneType</td>
<td>jason d williams florham_park new jersey. Say office, cell, or cancel.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action Search</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Values at point 51 (distance 0.028)</td>
<td></td>
</tr>
<tr>
<td>18.511 AskPhoneType</td>
<td></td>
</tr>
<tr>
<td>17.806 ConfirmPhoneType</td>
<td></td>
</tr>
<tr>
<td>17.546 AskName</td>
<td></td>
</tr>
</tbody>
</table>

Output system action
jason d williams florham_park new jersey. Say office, cell, or cancel.
### Reinforcement Learning: results

<table>
<thead>
<tr>
<th>Domain</th>
<th>Task completion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline</strong></td>
<td><strong>RL</strong></td>
</tr>
<tr>
<td>[1] Singh et al, 2002</td>
<td>20-64%</td>
</tr>
<tr>
<td>[2] Lemon et al, 2006</td>
<td>68%</td>
</tr>
<tr>
<td>[3] Frampton &amp; Lemon, 2008</td>
<td>82%</td>
</tr>
<tr>
<td>[4] Young et al, 2009</td>
<td>64%</td>
</tr>
<tr>
<td>[5] Thomson &amp; Young, 2009</td>
<td>84%</td>
</tr>
<tr>
<td>[6] Cuayahuitl et al, 2010</td>
<td>94%</td>
</tr>
</tbody>
</table>

---


---

*Slide taken from Williams, invited talk at ASRU 2009*
Related work on statistical dialogue management

- Large amount of work in slot-filling domains, e.g., flight reservations, restaurant recommendations (Schatzmann et al., 2006; Williams and Young, 2007; Janarthanam and Lemon, 2010; Rieser et al., 2011)
- Some work in natural language tutoring domains (Ai and Litman, 2007; Tetreault and Litman, 2008; Chi et al., 2010; Ai and Litman, 2011)
- Limited amount of work in negotiation domains (Heeman, 2009; Georgila and Traum, 2011a, 2011b)
User simulation evaluation metrics

For each system action in the corpus, we compare the actions of the real user with the most probable actions of the simulated user.

Precision = \frac{\text{correctly guessed actions}}{\text{all guessed actions}}

Recall = \frac{\text{correctly guessed actions}}{\text{all actions performed by real user}}

F-score balances precision and recall.

Example:

real user action: provide_hp, social_polite
simulated user action: provide_hp

Precision = 100% \quad Recall = 50\% \quad F\text{-score}=67\%

Georgila, Henderson and Lemon, 2005; 2006

Schatzmann, Georgila and Young, 2005
User simulation evaluation metrics (cont.)

- Compare real and simulated dialogues in terms of numbers of system and user actions, numbers of individual speech acts, grounding, formalities, etc. (Schatzmann et al., 2005)
- Kullback-Leibler divergence (Cuayáhuitl et al., 2005)
- More advanced simulated users train better dialogue policies (Schatzmann et al., 2005)
- When not much data is available hand-crafting the probabilities of simulated user actions is preferable (Ai & Litman, 2009)
Validation of simulated users

- Current metrics used for evaluating simulated users are not validated (i.e., are they behaving like real users?)

- Results from building simulated users and learning dialogue policies for older users agree with literature on diversity of older people’s behavior (Rabbitt and Anderson 2006)
  - Simulated users trained on older users cover the behavior of younger users but not the opposite
  - Simulated older users train better flexible dialogue strategies

- This is partial validation of the user simulation methodology

*Georgila, Wolters and Moore, 2008; 2010*
Overview

- Statistical approaches to dialogue management
  - Reinforcement learning
  - User simulation
  - User simulation evaluation metrics

- Speech synthesis for dialogue systems
  - Unit-selection speech synthesis
  - HMM speech synthesis
  - Expressive conversational speech synthesis
  - Amazon Mechanical Turk evaluation study
Speech synthesis (text-to-speech)

- Speech synthesis is the automatic conversion of text into speech
- Preprocessing stage
  - Sentence splitting
  - Part-of-speech tagging
  - Word-sense disambiguation
  - Parsing
  - Prosody prediction
- Realization stage
Unit-selection speech synthesis

- Small sub-word units of speech, e.g., phonemes or diphones, are recorded under high-quality recording conditions in as many contexts as possible, forming a database of natural speech.
- At runtime a synthetic utterance is created by putting together the best sequence of units from the database.
- The larger the database the better the quality of the synthetic voice.
- A limited-domain unit-selection voice is built using utterances from the domain in which the voice will be deployed.

Example problematic synthetic utterance:
Always respect your elders, that’s what I was taught.

Hunt and Black, 1996
### Unit-selection speech synthesis (cont.)

<table>
<thead>
<tr>
<th>Example target unit $t_i$</th>
<th>Example database unit $u_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type=r ih</td>
<td>Type=r ey</td>
</tr>
<tr>
<td>Pitch=120</td>
<td>Pitch=125</td>
</tr>
<tr>
<td>Duration=200ms</td>
<td>Duration=180ms</td>
</tr>
<tr>
<td>Preceding phoneme=vowel</td>
<td>Preceding phoneme=vowel</td>
</tr>
<tr>
<td>Following phoneme=consonant</td>
<td>Following phoneme=vowel</td>
</tr>
</tbody>
</table>

- Each target utterance (utterance to be synthesized) is a sequence of target units determined in the preprocessing stage.
- **Target cost** $C^t(u_i, t_i)$ is an estimate of the difference between a database unit $u_i$ and the target unit $t_i$.
- **Concatenation cost** $C^c(u_{i-1}, u_i)$ is an estimate of the quality of a join between consecutive units $u_{i-1}$ and $u_i$.
- The goal is to select the best sequence of database units to minimize the total cost (sum of target and concatenation cost).
HMM speech synthesis

- Hidden Markov Models (HMMs) are statistical models that model the average of some sets of similarly sounding speech units
- The idea is similar to using HMMs for speech recognition
- HMM speech synthesis is currently the most active area of research in speech synthesis
- HMM-based speech synthesis system (HTS)
  - http://hts.sp.nitech.ac.jp/

Zen, Tokuda and Black, 2009
Speaker-adaptive HMM speech synthesis

Depending on the type of the target speaker’s utterances, we can build voices for emotional or conversational speech, and for simulating foreign accents.

Utterances from various speakers (hours) ➔ Generic HMM voice ➔ HMM adaptation ➔ Utterances from target speaker (3-6 min) ➔ Target speaker HMM voice

The pipeline in the green box is built once only and reused for each target speaker.

Example synthetic utterance using speaker-adaptive HMM speech synthesis

*This is a synthetic utterance using statistical modeling of speech.*

*Yamagishi, Nose, Zen, Ling, Toda, Tokuda, King and Renals, 2009*
Unit-selection vs. HMM speech synthesis

- Example sentence: *I was about to do this when cooler judgment prevailed.*
  - Human speech
  - Unit-selection speech synthesis
  - HMM speech synthesis
Motivation for expressive conversational speech synthesis

- Virtual humans (and other types of dialogue systems) must be able to sound human-like and give the impression that they are engaged in the conversation.
- For example, they need to be able to handle speech disfluencies, e.g., filled pauses (*uh, um*) and fillers (*you know, well, yeah, I mean*).
The problem of generating speech disfluencies

- It is not enough to insert the disfluencies into the output of the natural language generation module but also need to synthesize them.
- Current state-of-the-art speech synthesizers
  - reach high levels of naturalness and intelligibility for read speech
  - **BUT**
  - **cannot** handle conversational speech

Example:
state-of-the-art commercial voice synthesizing read speech
*And then it's a big deal.*
state-of-the-art commercial voice synthesizing conversational speech
*Um yeah and then it's like a big deal, you know, right, so.*

- It is very important that we build speech synthesis systems capable of handling conversational speech in order to make our virtual agents more human-like.
Experiment on synthesizing speech disfluencies

- We use the CereProc speech synthesis system based on Unit Selection
  - Best speech synthesizer in our recent Amazon Mechanical Turk experiment
- Given plain text
  - Example
    - We wrote it that way to make it more interesting.
  - Add disfluencies in a consistent manner using a novel statistical algorithm based on n-grams and the Viterbi algorithm
    - Good example
      - We wrote it that way you know to to make it um more interesting.
    - Bad example
      - We wrote just it that way so to make um it more interesting.
- Synthesize the disfluent sentence

  Andersson, Georgila, Traum, Aylett and Clark, 2010
input:  <s> it’s a miracle </s>
Filler insertion algorithm

input:  <s> it’s a miracle </s>
output: <s> so it’s a miracle um yeah </s>
Training process

- We collected approx. 7 hours of spontaneous speech
  - by interviewing an actor and prompting him to speak about himself and his experiences
- We transcribed and cleaned up approx. 1 hour of data
- Using the cleaned up data
  - we trained the algorithm for inserting disfluencies
- We blended the cleaned up conversational data with read aloud data to build a new voice for conversational speech

Andersson, Georgila, Traum, Aylett and Clark, 2010
Evaluation

- Web-based listening test with 30 volunteering participants
- 2 test conditions
  - predicted fillers + spontaneous voice vs. predicted fillers + read voice
  - predicted fillers + spontaneous voice vs. no fillers + read voice
- 15 sentences for the two conditions in randomized order and also mirrored, thus 60 comparisons in total
- Participants were asked about their opinion on two different aspects
  - Conversational aspect
    - Which utterance in the pair sounds more like in an everyday conversation (as opposed to e.g., someone reading from a script)?
  - Naturalness
    - Which utterance in the pair sounds more natural (regardless if it sounds conversational or not)?
Examples of synthesized speech

Plain text: *I think it's out of insecurity but it's what I do.*

Text with inserted disfluencies: *I think it’s out of insecurity but yeah, it’s what I do.*

- Test condition 1
  - predicted fillers + spontaneous voice *vs.* predicted fillers + read voice

Plain text: *I like Mexican food.*

Text with inserted disfluencies: *Um I like Mexican food.*

- Test condition 2
  - predicted fillers + spontaneous voice *vs.* no fillers + read voice
Results of perceptual listening test

Spontaneous voice with fillers significantly more conversational than read voice with fillers

Spontaneous voice with fillers significantly more natural than read voice with fillers

Spontaneous voice with fillers significantly more conversational than read voice without fillers

Spontaneous voice with fillers and read voice without fillers not significantly different in terms of naturalness

*Andersson, Georgila, Traum, Aylett and Clark, 2010*
Conversational HMM speech synthesis

Fillers (Spontaneous HMM) vs. Fillers (Read aloud HMM)

Andersson, Yamagishi and Clark, 2012
Related work on synthesis/generation of speech disfluencies

  - spontaneous lecture monologues
  - model insertion and realization of filled pauses and breathing
  - limited domain speech synthesis

- Adell et al. (2006, 2007)
  - read aloud acted filled pauses
  - model insertion and realization of filled pauses
  - always synthesized filled pauses surrounded by silent pauses

- Campbell (2007)
  - corpus of spontaneous speech
  - phrase level concatenation
Virtual humans and speech output

- Virtual humans need to sound natural and give the impression that they are engaged in the conversation
- Question: use human recordings or synthesized speech?
  - What is the best trade-off between performance and cost?
- To address this question we perform two evaluation experiments (pilot study and Amazon Mechanical Turk study)
  - Human raters are asked to listen to utterances (human recordings and synthesized speech) and answer questions related to the quality of these utterances
  - We use 2 human voices, 2 general-purpose synthesized voices, and 2 limited-domain synthesized voices

Georgila, Black, Sagae and Traum, 2012
Domain - Simcoach

Simcoach aims to motivate military personnel and family members to seek information and advice about depression and post-traumatic stress disorder.
Corpus of sentences to be recorded or synthesized

- 200 sentences in the Simcoach domain (in-domain)
  - Example: *Well I’m just trying to get some info so that I can help you better.*

- 30 out-of-domain sentences
  - Example: *This TV show is hilarious, don’t you think so?*

- Long (>5 words) vs. short (<=5 words) sentences
  - Long in-domain example: *Now how about we go back to what we were talking about before?*
  - Short in-domain example: *Thanks for serving.*
  - Long out-of-domain example: *We can meet around 8 tonight if this is okay with you.*
  - Short out-of-domain example: *Nobody will notice.*
Variation in content

- **Positive long** example: *First thing you should know is that you’re not alone in this.*
- **Positive short** example: *Good work.*
- **Neutral long** example: *Would you like to exit and end this session?*
- **Neutral short** example: *Why’s that?*
- **Negative long** example: *It makes me feel unhappy and hopeless.*
- **Negative short** example: *They are disgusting.*

- Inter-annotator agreement for content
  - 83% agreement in terms of content (two annotators, 400 sentences)
  - For the two studies we selected only sentences where there was agreement between the annotators
Human and synthesized voices

Example sentence: *Feeling pretty good! More importantly, how are you?*

- Professional actor’s voice (PROF) – 100 in-domain sentences were recorded

- Amateur person’s voice (AMAT) – all 230 sentences were recorded
  - Due to recording problems (low audio volume) we only kept 67 sentences
  - Example problematic recording: *Guess we’re not a good fit.*

- High-quality general-purpose voice (GEN-HIGH) – all 230 sentences were synthesized
  - Custom voice developed by CereProc Ltd for USC/ICT
  - Unit-selection
Example sentence: *Feeling pretty good! More importantly, how are you?*

- **First limited-domain voice (LD1)** – all 230 sentences were synthesized
  - Developed using only in-domain sentences recorded by a professional actor; compatible with Flite

- **Second limited-domain voice (LD2)** – all 230 sentences were synthesized
  - Developed using in-domain sentences recorded by a speech synthesis expert plus some general material; compatible with Flite

- **Lower-quality general-purpose voice (GEN-LOW)** – all 230 sentences were synthesized
  - Microsoft Sam
Experimental setup – Pilot study

Each participant:
- listen to 12 utterances (2 each)
  - PROF
  - AMAT
  - LD1
  - LD2
  - GEN-HIGH
  - GEN-LOW
- answer 3 questions
  - Naturalness
  - Conversational aspect
  - Likability
Results – Pilot study (only positive-content utterances were used)

27 participants rated 324 utterances; no significant differences between PROF and AMAT; no significant differences between GEN-HIGH, LD1, and LD2
Experimental setup – Amazon Mechanical Turk study

- Same questions as for pilot study except for the question that measures conversational aspect which changed to
  - Does this utterance sound more like in an everyday conversation (as opposed to e.g., someone reading from a script)?
    - 1=definitely not like in an everyday conversation, 2=perhaps not like in an everyday conversation, 3=cannot decide, 4=perhaps like in an everyday conversation, 5=definitely like in an everyday conversation
- Each participant could perform one or more HITs (Human Intelligence Tasks) and each HIT contained 5 utterances
  - To prevent spam HITs, if the 5 sentences of the HIT were all done in less than 5 seconds, then all the submissions in that HIT were discarded
Results – Amazon Mechanical Turk study (utterances with all types of content were used)

826 participants rated 24590 utterances; overall PROF is the best (significant); GEN-HIGH and LD1 better than AMAT and LD2 (significant only for all questions combined)
Summary of results – Amazon Mechanical Turk

- PROF supersedes all voices (significant)
- GEN-HIGH is rated higher than AMAT (significant for all questions combined, naturalness, and likability)
- LD1 is rated higher than AMAT (significant only for all questions combined)
- No significant difference between GEN-HIGH and LD1, except for naturalness where GEN-HIGH is rated higher than LD1
- LD1 is rated higher than LD2 (significant for all questions combined, naturalness, and likability, but not for conversational aspect or different variations depending on domain, length, and content)
- Generally GEN-HIGH is rated higher than LD1 for out-of-domain sentences and lower for in-domain sentences
- There is a trend for long and negative content sentences to receive lower ratings (not significant)
- No useful conclusions regarding the impact of the raters’ native language, age, and familiarity with speech synthesis on their ratings

Georgila, Black, Sagae and Traum, 2012
Thank you

Thanks very much!

kgeorgila@ict.usc.edu
http://people.ict.usc.edu/~kgeorgila/