Harnessing Speech Prosody for Human-Computer Interaction

Elizabeth Shriberg       Andreas Stolcke

Speech Technology and Research Laboratory
SRI International, Menlo Park, CA

International Computer Science Institute
Berkeley, CA
Collaborators

- Lee Stone (NASA); Beth Ann Hockey, John Dowding, Jim Hieronymous (RIACS)
- Luciana Ferrer (SRI postdoc)
- Harry Bratt, Kemal Sonmez (SRI)
- Jeremy Ang (ICSI/UC Berkeley)
- Emotion labelers: Raj Dhillon, Ashley Krupski, Kai Filion, Mercedes Carter, Kattya Baltodano
Introduction

- **Prosody** = melody, rhythm, “tone” of speech
- Not *what* words are said, but *how* they are said
- Human languages use prosody to convey:
  - phrasing and structure (e.g. sentence boundaries)
  - disfluencies (e.g. false starts, repairs, fillers)
  - sentence mode (statement vs question)
  - emotional attitudes (urgency, surprise, anger)
- Currently largely unused in speech systems
Talk Outline

- Project goal and impact for NASA
- Sample research tasks:
  - Task 1: **Endpointing**
  - Task 2: **Emotion classification**
- General method
  - language model, prosodic model, combination
  - data and annotations
- Results
  - **Endpointing**: error trade-offs & user waiting time
  - **Emotion**: error trade-offs & class definition effects
- Conclusions and future directions
Project Goal

- Most dialog systems don’t use prosody in input; they view speech simply as “noisy” text.

- Our goal: add prosodic information to system input.
Today: Two Sample Tasks

- Task 1: Endpointing (detecting end of input)
  - current ASR systems rely on pause duration
  - measure temperature at . . . cargo bay . . .
  - causes premature cut-off during hesitations
  - wastes time waiting after actual boundaries

- Task 2: Emotion detection
  - word transcripts don’t indicate user state
  - measure the -- STOP!! GO BACK!!
  - alert computer to immediately change course
  - alert other humans to danger, fatigue, etc.
Other Tasks in Project

- Automatic sentence punctuation:
  - *Don’t go to flight deck!*
  - *Don’t! Go to flight deck!* (DO go to flight deck)

- Detection of utterance mode:
  - Computer: Confirm opening of hatch number 2
  - Human: *Number 2.* /? (confirmation or question?)

- Detection of disfluencies:
  - *Item three one five one two* (item 31512 or 512?)
Method: Prosodic Modeling

- Pitch is extracted from acoustic signal
- Speech recognizer identifies phones, words, and their durations
- Pitch and duration information is combined to compute distinctive *prosodic features* (e.g., Was there a pitch fall/rise in last word?)
- *Decision trees* are trained to detect desired events from features
- Separate test set used to evaluate classifier performance
Method: Language Models

- Words can also predict events of interest, using N-gram language models.
- Endpointing -- predict endpoint probability from last two words:  \( P(\text{endpoint} \mid \text{word}_{-1}, \text{word}_{-2}) \)
- Emotion detection -- predict from all words in sentence:  \( P(\text{word}_1, \text{word}_2, \ldots, \text{word}_n \mid \text{emotion}) \)
- \( P > \text{threshold} \Rightarrow \text{system detects event} \)
- Prosodic classifier and LM predictions can be combined for better results (multiply predictions)
Task 1: Endpointing in ATIS

- **Air Travel Information System** = Dialog task defined by DARPA to drive research in spoken dialog systems
- Users talk to a (simulated) air travel system
- Simulated endpointing “after the fact”
- About 18,000 utterances, 10 words/utterance
- Test set of 1974 unseen utterances
- 5.9% word error rate on test set
Endpointing Algorithms

- **Baseline algorithm:**
  - Pick pause threshold for decision
  - Detect endpoint when pause duration > threshold

- **Endpointing with prosody and/or LM:**
  - Pick probability threshold for decision
  - Train separate classifiers for pause values > .03, .06, .09, .12, .25, .50, .80 seconds
  - For each pause threshold:
    - Detect endpoint if classifiers predicts probability > threshold
    - Otherwise wait until next higher pause threshold is reached
  - Detect endpoint when pause > 1s
Endpointing Metrics

- **Performance metrics:**
  - False alarms: system detects false endpoint
  - Misses: system fails to detect true endpoint
  - Recall = % of true endpoints detected  
    = 1 – Miss rate

- **Error trade-off**
  - System can be set more or less "trigger-happy"
  - Fewer false negatives ↔ More false positives
  - Equal error rate (EER): error rate at which false alarms = misses

- **ROC curve:** graphs recall vs. false alarms
ATIS Endpointing Results

- Endpointer used automatic recognition output (5.9% WER. Note: LM degrades with WER).

- Equal Error Rates
  - Baseline: pause threshold 8.9 %
  - Prosodic decision tree only 6.7 %
  - Language model only 6.0 %
  - Prosody + LM combined 5.3 %

- Prosody alone beats baseline
- Combined classifier better than LM alone
ROC for Endpointing in ATIS
ATIS Examples

- Do you have a flight between Philadelphia and San Francisco?
  - Baseline makes false endpoints at the locations (so would cut speaker off prematurely)
  - Prosody model waits, despite the pause, because pitch doesn’t move much, stays high (hesitation)

- I would like to find the cheapest one-way fare from Philadelphia to Denver.
  - Prosody mistakenly predicts endpoint (“?” rise)
  - Combined prosody and LM endpointer avoids false endpoint (rare to end on “cheapest”).
Prosodic Cues for Endpointing

- Pitch range
  - speaker close to his/her estimated F0 “baseline” or “topline” (logratio of fitted F0 in previous word to that measure)
  - baseline/topline estimated by LTM model of pitch

- Phone and syllable durations
  - last vowel or syllable rhyme is extended
  - normalized for both the segmental content (intrinsic duration) and the speaker
Endpointing Speed-Up

- **User Waiting Time** = average pause delay needed for system to detect true endpoints

- In addition to preventing false alarms, prosody reduces UWT for any given false alarm rate:

<table>
<thead>
<tr>
<th>False Alarms</th>
<th>2%</th>
<th>4%</th>
<th>6%</th>
<th>8%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>.87</td>
<td>.54</td>
<td>.38</td>
<td>.26</td>
<td>.15</td>
</tr>
<tr>
<td>Tree only</td>
<td>.82</td>
<td>.32</td>
<td>.18</td>
<td>.10</td>
<td>.09</td>
</tr>
<tr>
<td>Tree + LM</td>
<td>.69</td>
<td>.23</td>
<td>.10</td>
<td>.06</td>
<td>.05</td>
</tr>
</tbody>
</table>

- Result: zippier interaction with system
Endpointing in a NASA Domain

- Personal Satellite Assistant: Dialog system controlling a (simulated) on-board robot
- Developed at NASA Ames/RIACS
- Data courtesy of Beth Ann Hockey
- Endpointer trained on ATIS, tested on 3200 utterances recorded at RIACS
- Used transcribed words
- "Blind test": no training on PSA data!
Endpointing in PSA Data

- ATIS language model not applicable, not used for endpointing
- PSA data had no utterance-internal pauses ⇒ baseline and prosodic model had same EER = 3.1% (no opportunity for false alarms)
- However: prosody still saves time:
  UWT (in seconds) at 2% false positive rate
  - Baseline 0.170
  - Prosodic tree 0.135
- Prosodic model is portable to new domains!
PSA Example

- Move to commander’s seat and measure radiation

- Baseline and prosody system both configured (via decision thresh.) for 2% false alarm rate

- As noted earlier, no error diffs for this corpus

- But baseline system takes 0.17s to endpoint after last word.

- Prosody system takes only 0.04s to endpoint!
Task 2: Emotion Detection

- Issue of data: used corpus of HC telephone dialogs labeled for emotion for DARPA project
- Would like more realistic data, with fear, etc.
- DARPA data: main emotion = frustration
- Each dialog labeled by 2+ people independently
- 2nd “Consensus” pass for all disagreements, by two of the same labelers.
Labeled Classes

- **Emotion**: neutral, annoyed, frustrated, tired/disappointed, amused/surprised,

- **Speaking style**: hyperarticulation, perceived pausing between words or syllables, raised voice

- **Repeats and corrections**: repeat/rephrase, repeat/rephrase with correction, correction only

- **Miscellaneous useful events**: self-talk, noise, non-native speaker, speaker switches, etc.
Emotion Samples

- **Neutral**
  - July 30
  - Yes

- **Disappointed/tired**
  - No

- **Amused/surprised**
  - No

- **Annoyed**
  - Yes
  - Late morning (HYP)

- **Frustrated**
  - Yes
  - No
  - No, I am … (HYP)
  - There is no Manila...
### Results: Annoy/Frust vs All Others

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (%)</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(chance = 50%)</td>
<td>(Acc-C)/(1-C)</td>
</tr>
<tr>
<td>Each Human with Other Human, overall</td>
<td>71.7</td>
<td>.38</td>
</tr>
<tr>
<td>Human with Human “Consensus” (biased)</td>
<td>84.2</td>
<td>.68</td>
</tr>
<tr>
<td>Prosodic Decision Tree with Consensus</td>
<td>75.6</td>
<td>.51</td>
</tr>
<tr>
<td>Tree with Consensus, no repeat/correction</td>
<td>72.9</td>
<td>.46</td>
</tr>
<tr>
<td>Tree with Consensus, repeat/correction only</td>
<td>68.7</td>
<td>.37</td>
</tr>
<tr>
<td>Language Model features only</td>
<td>63.8</td>
<td>.28</td>
</tr>
</tbody>
</table>
Results (cont.)

- H-H labels agree 72%, complex decision task
  - inherent continuum
  - speaker differences
  - relative vs. absolute judgements

- H labels agree 84% with “consensus” (biased)

- Tree model agrees 76% with consensus-- better than original labelers with each other

- Prosodic model makes use of a dialog state feature, but without it it’s still better than H-H

- Language model features alone are not good predictors (dialog feature alone is better)
Baseline Prosodic Tree

duration feature  pitch feature  other feature

REPCO in ec2,rr1,rr2,rex2,inc,ec1,rex1 :  0.7699 0.2301 AF
  MAXF0_IN_MAXV_N < 126.93:  0.4735 0.5265 N
  MAXF0_IN_MAXV_N >= 126.93:  0.8296 0.1704 AF
    MAXPHDUR_N < 1.6935:  0.6466 0.3534 AF
      UTTPOS < 5.5:  0.1724 0.8276 N
      UTTPOS >= 5.5:  0.7008 0.2992 AF
    MAXPHDUR_N >= 1.6935:  0.8852 0.1148 AF
REPCO in 0 :  0.3966 0.6034 N
  UTTPOS < 7.5:  0.1704 0.8296 N
  UTTPOS >= 7.5:  0.4658 0.5342 N
    VOWELDUR_DNORM_E_5 < 1.2396:  0.3771 0.6229 N
      MINF0TIME < 0.875:  0.2372 0.7628 N
      MINF0TIME >= 0.875:  0.5 0.5 AF
        SYLRATE < 4.7215:  0.562 0.438 AF
          MAXF0_TOPLN < -0.2177:  0.3942 0.6058 N
          MAXF0_TOPLN >= -0.2177:  0.6637 0.3363 AF
        SYLRATE >= 4.7215:  0.2816 0.7184 N
    VOWELDUR_DNORM_E_5 >= 1.2396:  0.5983 0.4017 AF
      MAXPHDUR_N < 1.5395:  0.3841 0.6159 N
        MINF0TIME < 0.435:  0.1 0.9 N
        MINF0TIME >= 0.435:  0.4545 0.5455 N
          RISERATIO_DNORM_E_5 < 0.69872:  0.3284 0.6716 N
          RISERATIO_DNORM_E_5 >= 0.69872:  0.6111 0.3889 AF
      MAXPHDUR_N >= 1.5395:  0.6728 0.3272 AF
Predictors of Annoyed/Frustrated

- Prosodic: Pitch features:
  - high maximum fitted F0 in longest normalized vowel
  - high speaker-norm. (1st 5 utts) ratio of F0 rises/falls
  - maximum F0 close to speaker’s estimated F0 “topline”
  - minimum fitted F0 late in utterance (no “?” intonation)

- Prosodic: Duration and speaking rate features
  - long maximum phone-normalized phone duration
  - long max phone- & speaker- norm.(1st 5 utts) vowel
  - low syllable-rate (slower speech)

- Other:
  - utterance is repeat, rephrase, explicit correction
  - utterance is after 5-7th in dialog
## Effect of Class Definition

<table>
<thead>
<tr>
<th>Baseline prosody model Consensus labels</th>
<th>Accuracy (%) (chance = 50%)</th>
<th>Entropy Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>A,F vs. N,else</td>
<td>75.6</td>
<td>21.6</td>
</tr>
<tr>
<td>Tokens on which labelers originally agreed</td>
<td>78.3</td>
<td>26.4</td>
</tr>
<tr>
<td>A,F vs. N,else</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All tokens Consensus labels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F vs. A,N,else</td>
<td>82.7</td>
<td>37.0</td>
</tr>
</tbody>
</table>

For **less ambiguous or more extreme** tokens, performance is significantly better than our baseline.
Error trade-offs (ROC)
Results Summary

- Prosody allows significantly more accurate (fewer false cut-offs) and faster endpointing in spoken input to dialog systems.
- Prosodic endpointer is portable to new applications. (Note: language model is not!)
- Prosody significantly improves detection of frustration over (cheating) language model.
- Prosody is of further value when combined with lexical information, regardless of which model is better on its own.
Impact and Future Work

- Prosody enables more accurate spoken language processing by capturing information “beyond the words”.
- Prosody creates new capabilities for systems (e.g., emotion detection).
- Prosody can speed up HCI (e.g., endpointing).
- Prosody presents potential for fusion with other communication modalities, such as vision.
Thank You