Learning from Mistakes: Expanding Pronunciation Lexicons using Word Recognition Errors

Sravana Reddy
The University of Chicago

Joint Work with Evandro Gouvêa
Sang Bissenette

SPEECH RECOGNITION

Sane visitor
Mariano DiFabio

SPEECH RECOGNITION

Mary and the fable
This Work

Mariano DiFabio

Out of Vocabulary (OOV) Words

Black Box

Latent Phonetic Similarity Channel

Pronunciations of OOV words (Mariano and DiFabio)

Known Words

Mary and the fable
Previous Work

Mariano DiFabio

Mary and the fable

Pronunciations of OOV words (Mariano and DiFabio)
Previous Work

- Wooters and Stolcke (ICASSP 1994)
- Sloboda and Waibel (ICSLP 1996)
- Fossler-Lussier (Ph.D. Thesis 1999)
- Maison (Eurospeech 2003)
- Tan and Bessacier (Interspeech 2008)
- Bansal et al (ICASSP 2009)
- Badr et al (Interspeech 2010)
  etc.
Why assume black-box access?

- **Practical:** What if ASR engine is a black box? (proprietary speech recognition tools, etc.)

  - *Example possible use of our approach:* Third-party app analyzes results of black-box recognition engine, returns OOV pronunciations

- **Scientific:** How much pronunciation information can we get from only word recognition errors?
Our Generative Model…

… for input word $w$ and output recognition hypothesis $e$

1. Generate word $w$ with $\text{Pr}(w)$

2. Generate pronunciation baseform $b$ with $\text{Pr}(b|w)$

3. Generate phoneme sequence $p$ with $\text{Pr}(p|b,w)$ by passing through phonetic confusion channel

4. Generate hypothesis word or phrase $e$ with $\text{Pr}(e|p,b,w)$

$$\text{Pr}(w,e) = \sum_{b,p} \text{Pr}(w) \text{Pr}(b|w) \text{Pr}(p|b,w) \text{Pr}(e|p,b,w)$$
Our Generative Model...

… for input word w and output recognition hypothesis e

1. Generate word w with Pr(w)

2. Generate pronunciation baseform b with Pr(b|w)

3. Generate phoneme sequence p with Pr(p|b, w) by passing through phonetic confusion channel

4. Generate hypothesis word or phrase e with Pr(e|p, b, w)

5. Repeat steps 2-4 to generate more e
Learning Algorithm

**GOAL**: find best pronunciation for input word \( w \)

\[
\operatorname*{arg\ max}_{b} \Pr(b \mid w)
\]

**Given**

- Current guess about \( \Pr(\text{baseform } b \mid w) \)
- \( \Pr(\text{transformed phonemes } p \mid b, w) \)  -- will explain later
- \( \Pr(\text{word recognition output } e \mid p, b, w) = \Pr(e \mid p) \)  Current Lexicon

**Phonetic Confusions**
Learning Algorithm

- **Compute posterior probability of baseform** \( b \) **given** \( w \) **and** \( e \)

\[
Pr(b \mid e, w) = \sum_c \frac{Pr(b \mid w) Pr(p \mid b, w) Pr(e \mid p, b, w)}{Pr(c \mid w) Pr(p \mid c, w) Pr(e \mid p, c, w)}
\]

- **Sum over all** \( e \) **in** \( n \)-best word recognition lists **over all utterances of** \( w \)

\[
Pr(b \mid w) = \sum_{e \in E_w} Pr(b \mid e, w) Pr(e)
\]
Initial Guess for $\Pr(b \mid w)$

- Limit to reasonable candidates

$\Pr(b \mid w) = \frac{1}{|B_w|}$ if $b \in B_w$

$0$ otherwise

* Bisani and Ney (2008)
Modeling Phonetic Confusions

\[ \Pr(p|b,w) = \Pr(p|b) = \text{sum of paths with input } b \text{ & output } p \]
Data

- CSLU Names Corpus
- Only use single-word names (isolated-word experiments)
- 20423 utterances, 7771 unique names

- *Train* (learn OOV pronunciations):
  Random 50% of utterances for each name

- *Test* (evaluate new lexicon):
  Remaining utterances
Setup

- Sphinx 3
- MFCCs extracted using Sphinx’s default parameters
- Acoustic Models trained on TIMIT
- Original Lexicon: CMU Dictionary, CSLU names removed
- Language Model: unigrams over names, add-one smoothing to include all CMU Dictionary words
Evaluation

- **Word Error Rate** of ASR recognition with learned lexicon

- **Baseform Error Rate**: proportion of learned baseforms different from corpus transcriptions

- **Phoneme Error Rate**: proportion of insertions, deletions, and substitutions of learned baseforms against corpus transcriptions

**Baselines:**

1. State of the art g2p: Sequitur, multigrams of order 6 *(SEQUITUR)*
2. CMU Dictionary pronunciations for names in dictionary *(CMuGOLD)*
Can we get better pronunciations than a grapheme-to-phoneme system?
Results

(Only those utterances where the names are in the CMU Dictionary)

E_w (set of hypotheses) = results from 10-best recognition

E_w = results from 5-best recognition

CMUGOLD

Word Error Rate of ASR recognition with learned lexicon

How does ASR recognition with gold standard pronunciations compare?
Results

$E_w$ (set of hypotheses) = results from 10-best recognition

$E_w = \text{results from 5-best recognition}$

SEQUITUR

<table>
<thead>
<tr>
<th>Baseform Error Rate against manual transcriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>25</td>
</tr>
</tbody>
</table>
Results

$E_w$ (set of hypotheses) = results from 10-best recognition

$E_w = \text{results from 5-best recognition}$

SEQUITUR

Phoneme Error Rate against manual transcriptions
What Works?

Dense phonetic neighborhood

- Merry in
- Mary
- Mary and
- Marian
- Marilyn
- Perelman
- Maritime

Successful pronunciation recovery

Sparse phonetic neighborhood

- Rumor for
- Rutherford
- Luther of
- Ruder for

Not so successful
Conclusion

- Can we **learn pronunciations from word recognition errors**?
  - Yes!
  - Learned pronunciations are better than grapheme-to-phoneme results

- Preliminary work – lots more to be done
  - Extend EM to also learn (or augment) phonetic confusions
  - Learn pronunciation variants of words in lexicon
  - Adapt to continuous speech (not just isolated words)
  - Seed \(P(b|w)\) independent of Sequitur or other g2p
  - Combine phone lattice information and word recognition output as cues for pronunciation
Dank Yu!