Detection of Lexical Stress Using an Iterative Feature Normalization Method

Erez Oxman, Eduard Golshtein
Word Stress

• Word stress means putting emphasis on a syllable in a word

Why is stress important?
• Distinguish between words:
  Subject vs. Subject

• Understandability
Lexical Stress Features

- Observed features:
  - $E =$ Energy
  - $D =$ Duration
  - $P =$ Pitch ($f_0$)

Stress is relative:

$$Ratios: \frac{E_{AE}}{E_{IH}}, \frac{D_{AE}}{D_{IH}}, \frac{P_{AE}}{P_{IH}}$$
Feature Variability

Observed features are highly variable:

- **Intrinsic Phrase Features**
  - Speaking rate
  - Speech intensity

- **Intrinsic Phoneme Features**
  - **Phoneme type** – Low vowels more energetic than high vowels.
  - **Phoneme stress** – Primary stress is more energetic than Unstressed
  - **Phoneme location** – Intensity stronger at beginning than at end.

Solution - Normalization

How to calculate?

Too many…
How to calculate the normalization parameters?

- Measure directly?
  \[ \mu_{AE} = \frac{1}{N_{AE}} \sum E_{AE} \]

- Measure Ratios?
  \[ \frac{\mu_{AE}}{\mu_{IH}} = \frac{1}{N_{AE,IH}} \sum \frac{E_{AE}}{E_{IH}} \]

Ratios between observed features are meaningless unless taken from the same phrase. Summation has to go over phrases that have both IH,AE. Very large database for all possible pairs.

Iterative method:

\[ \mu_{AE}^{1+1} = \frac{1}{N_{AE}} \sum \tilde{E}_{AE} \left( \frac{\mu_{AE}}{\mu_{IH}} \right) \]

Where:

\[ \tilde{E}_{AE} = \frac{E_{AE}}{1 \left( \frac{E_{AE}}{\mu_{AE}} + \frac{E_{IH}}{\mu_{IH}} \right)} \]

Do the normalization parameters converge?
Do they converge to the average of the intrinsic phoneme energies?
Simulation details

- 40,000 words.
- 9 different phonemes with different intrinsic phoneme averages $\mu_1 = 1, \mu_2 = 2, ... \mu_9 = 9$
- Observed syllable energies were created using this formula:

$$E_{\text{Syllable}} = M_{\text{Phrase}} \ast e_{\text{IntrinsicSyllableEnergy}}$$

$$N(\mu_i, \sigma_i = 0.1\mu_i)$$

Random number
Convergence

Feature Distribution

Before Iterations

After Iterations
Experimental Description
Partition to Classes

• Class = specific set of phonemes, stress levels, locations... e.g. \( \{\mu_{AE}, \mu_{IH}, \ldots\} \)

• The following partitions were used in our experiments:
  - **Phoneme type**: High vowels, Mid vowels, Low vowels, Dipthong, Syllabic Consonants
  - **Phoneme stress level**: Primary, Secondary, Unstressed
  - **Phoneme location**: First phoneme, First syllable and not first phoneme, Last phoneme, Last syllable and not last phoneme
Training Algorithm

Given DB transcribed+segmented+stress:

1. Partition to classes with different intrinsic properties
2. Calculate \( \{\mu\} \) using iterations
3. Extract normalized features
4. Train 3 models – Primary, Secondary, Unstressed (GMM+EM)
Detection System - Disyllabic

\[ \tilde{E}_{AE} = \frac{E_{AE}}{1 \pm \frac{1}{2} \left( \frac{E_{AE}}{\mu_{AE,Pr}} + \frac{E_{IH}}{\mu_{IH,Un}} \right)} \]

Transcribed & Segmented Speech

\[ \mu_{AE,Pr} \neq \mu_{AE,Un} \]

Observed Features Calculation

Partition according to stress levels
# System Performance Detecting the Position of the Primary Stress

Tested relative to expert native transcribers

<table>
<thead>
<tr>
<th>System\Mother Tongue</th>
<th>Hebrew</th>
<th>Native English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference – Common normalization $\mu = 1$</td>
<td>92.1</td>
<td>96.6</td>
</tr>
<tr>
<td>12 Classes (Stress+Position)</td>
<td>92.4</td>
<td>96.6</td>
</tr>
<tr>
<td>60 Classes (Stress+Position+PhonType)</td>
<td>93.2</td>
<td>96.9</td>
</tr>
</tbody>
</table>

Hebrew speakers change pitch during syllables

| 60 Classes + Pitch gradient                       | 94.9         | 97.1           |

Feature’s contribution is language dependent

DB: Native English: Train: 20,000 disyllabic words, Test: 5,000 disyllabic words

Hebrew: Train: 12,000 disyllabic words, Test: 1,300 disyllabic words
### Performance: System vs. Listeners

4 native non-expert listeners per recording:

<table>
<thead>
<tr>
<th>Agreement between listeners</th>
<th>Percent of test</th>
<th>System’s Performance</th>
<th>Performance of “Majority Decision” of listeners</th>
<th>Average Performance of listeners</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>67%</td>
<td>96.4</td>
<td>99.7</td>
<td>99.7</td>
</tr>
<tr>
<td>75%</td>
<td>29%</td>
<td>92.3</td>
<td>95.5</td>
<td>72.8</td>
</tr>
<tr>
<td>50%</td>
<td>4%</td>
<td>87.2</td>
<td>50.0</td>
<td>50.0</td>
</tr>
<tr>
<td>-----</td>
<td>100%</td>
<td>94.9</td>
<td>96.7</td>
<td>90.1</td>
</tr>
</tbody>
</table>

System underperforms the performance of the majority decision of listeners

System outperforms the average performance of listeners
Conclusions

• New data-driven normalization technique that reduces the variability of the intrinsic phoneme features.

• Feature’s contribution is language dependent

• System outperforms the average performance of listeners
THANK YOU