Microsoft Research

Each year Microsoft Research hosts hundreds of influential speakers from around the world including leading scientists, renowned experts in technology, book authors, and leading academics, and makes videos of these lectures freely available.

2016 © Microsoft Corporation. All rights reserved.
Multilingual Modulation by Neural Language Codes

Markus Müller, Sebastian Stüker and Alex Waibel

Institute for Anthropomatics and Robotics, Interactive Systems Lab
Outline

- Introduction
- Language Codes
- Integration of language codes into the network architecture
- Phonetic pre-training
- Network superstructure (Meta-PI)
- Conclusion

Introduction

- Automatic speech recognition (ASR): Costly AI problem
  - 7,000+ living languages, each requires own acoustic model

- How to build a system for a language?
  - $L_x$ on EN (cross-lingual): worst performance
  - $L_1, L_2, \ldots L_n$ on EN (multilingual): mediocre performance
  - EN on EN (monolingual): best performance

- Monolingual setup wins

- Typical multilingual training
  - Train model on multiple languages
  - Fine-tune on target language

- Want: Quick adaptation to languages
  - Monolingual performance multilingually
Multilingual Ambiguity

- Asynchronous transition of articulators between phones
- Context-dependent coarticulation artifacts
  - e.g. shifts in tongue position endpoints
Multilingual Neural Network Adaptation

- Multilingual acoustic model: Multilingual set of acoustic units
  - IPA: Same symbols across languages, language specific contexts
  - Multilinguality adds more ambiguity, performance loss

- Adaptation method: Networks modulated by language codes
  - Extracted via ancillary network trained on auxiliary task

- Stimulate networks to learn features depending on extracted language properties

- Optimized neural network architecture and integration of language codes

- Achieved and exceeded parity with monolingual setups

- Instant adaptation to languages
Neural Network Language Adaptation

- Supply additional language code
- Language identity (LID)
  - One-hot encoding of identity
- Language Feature Vectors (LFV)
  - Encoding of language properties
  - Extracted via bottleneck layer

Acoustic feat. | LFV Bottleneck
---|---

LID Network | LFV:

LID: DE EN TR

X_1 X_2 X_n
Language Codes

- Input features: Context ± 33 frames (≈ 700ms)
  - Language properties: Longer-duration

- Output features: Smoothed on utterance basis
  - Difficult for online scenarios
  - Smoothing on speaker level also works

Language Feature Vectors (t-SNE)

- t-SNE projection of LFVs, colored by language identity
Language Feature Vectors (t-SNE)

- t-SNE projection of LFVs, colored by language identity

English
LC Analysis: Language Identification

- Computed LFVs on training data, averaged per language
  - Language prototype vectors

- Recorded test set and computed distance to prototype vectors
  - German speaker with strong accent reading English sentences
THE HIDDEN CODE OF LANGUAGE INDEPENDENCE
Language Codes

- Input features: Context +/− 33 frames (∼700ms)
  - Language properties: Longer-duration

- Output features: Smoothed on utterance basis
  - Difficult for online scenarios
  - Smoothing on speaker level also works

Neural Network Language Adaptation

- Supply additional language code
- Language identity (LID)
  - One-hot encoding of identity
- Language Feature Vectors (LFV)
  - Encoding of language properties
  - Extracted via bottleneck layer
Language Codes

- Input features: Context +/- 33 frames (≈ 700ms)
  - Language properties: Longer-duration

- Output features: Smoothed on utterance basis
  - Difficult for online scenarios
  - + Smoothing on speaker level also works

Comparison of BiLSTM/CTC Architectures

- Language properties not as signal related as speaker properties
- Integrate language adaptation deeper into the network
- Additive language codes
- Multiplicative language codes
Multiplicative Language Codes

- Neural network modulation related to modulation in Meta-PI
- Outputs weighted by language codes
  - Emphasized / attenuated based on language properties
  - Forces neural units to learn features depending on LCs
  - Network instantly adapts to languages
- Language codes reconfigure network based on language features
- “Intelligent” dropout, gate neural activations
Experimental Setup BiLSTM/CTC Systems

- Trained on 4 languages (English, French, German, Turkish)
  - 45h per language

- Audio front-end: Multilingual bottleneck features (ML-BNFs)
  - Trained on logMel and tonal features
  - 5 languages: French, German, Italian, Russian, Turkish

- No pronunciation dictionaries used
  - Trained on characters only
    - Network has to infer pronunciations automatically

- Character based RNN language model
  - Trained on 0.5 million words of training transcripts

- Evaluation metrics
  - CER, WER
Comparison Additive and Multiplicative Codes

- Multilingual systems, Character Error Rate (CER)
- Applying language codes deeper into the network improves performance

<table>
<thead>
<tr>
<th>Adaptation</th>
<th>(\emptyset)</th>
<th>ML</th>
<th>DE</th>
<th>EN</th>
<th>FR</th>
<th>TR</th>
</tr>
</thead>
<tbody>
<tr>
<td>No adaptation</td>
<td>14.4%</td>
<td>10.6%</td>
<td>18.2%</td>
<td>15.9%</td>
<td>9.1%</td>
<td></td>
</tr>
<tr>
<td>Additive</td>
<td>13.0%</td>
<td>9.5%</td>
<td>16.1%</td>
<td>14.3%</td>
<td>8.1%</td>
<td></td>
</tr>
<tr>
<td>Multiplicative</td>
<td>12.4%</td>
<td>9.1%</td>
<td>15.5%</td>
<td>13.6%</td>
<td>8.0%</td>
<td></td>
</tr>
</tbody>
</table>

Neural Network Stimulation

- Modulation with language codes enables language aware nets
  - Weights applied to connections reconfigure network
- Use explicitly modelled knowledge to learn feature detectors
- Proof-of-concept: include phonetic knowledge into the network
  - Use pronunciation dictionaries
  - Pre-train part of network for phone detection
  - Pre-train part of network for pronunciation modelling
- Jointly train assembled network
**Phonetic Pre-training**

- Include phonetic knowledge
- Global phoneme set
  - Language specific coloring
- Based on existing architecture
- Training schedule
  1. Pre-train on phonetic targets
  2. Freeze net, add another block → pre-condition weights
  3. Train whole net jointly

Results Phonetic Pre-training

- Pre-training the net lowers CER
- Contrastive experiment: more layers

<table>
<thead>
<tr>
<th>Setup</th>
<th>DE</th>
<th>EN</th>
<th>FR</th>
<th>TR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (4L)</td>
<td>7.7%</td>
<td>11.0%</td>
<td>8.8%</td>
<td>6.0%</td>
</tr>
<tr>
<td>Baseline (6L)</td>
<td>9.0%</td>
<td>12.7%</td>
<td>10.3%</td>
<td>7.5%</td>
</tr>
<tr>
<td>Pre-training (6L)</td>
<td>6.0%</td>
<td>10.0%</td>
<td>8.7%</td>
<td>5.3%</td>
</tr>
</tbody>
</table>

- Evaluation on English, WER

<table>
<thead>
<tr>
<th>Setup</th>
<th>EN CER</th>
<th>EN WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>11.0%</td>
<td>26.3%</td>
</tr>
<tr>
<td>Pre-training</td>
<td>10.0%</td>
<td>25.4%</td>
</tr>
</tbody>
</table>
Meta-PI Networks: Distributed Knowledge

Network Superstructure for Multilingual ASR

- Modulation (already covered)
  - Apply weights to outputs of neural units
- Train smaller subnets on individual tasks
  - Language dependent subnets
- Learn mixture weights of subnets based on final task
  - Train adaptive neural language codes (NLCs) based on LFVs
- Joint training of entire network superstructure
  - Parameters of individual networks updated
  - Monolingual subnets adapted to multilingual speech recognition
Network Architecture

- Stack outputs of subnets
  - Language dependent
  - Remove output layers
  - Stack outputs of last hidden layers

- Main network
  - 2 BiLSTM blocks

- Joint training of all networks
  - Update pre-trained language dependent networks
  - Update NLCs
Adaptive Neural Language Codes

- Match dimensionality of layer output and language code
  - Stack multiple instances
- Optimize language code for speech recognition
- Input: ML-BNFs and LC
- Net learns stacked LFV output
  - Adapts output to new task
Results

- Network superstructure and NLCs improve performance
  - Evaluation on English

<table>
<thead>
<tr>
<th>Setup</th>
<th>WER LM1</th>
<th>WER LM2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monolingual baseline</td>
<td>25.3%</td>
<td>24.2%</td>
</tr>
<tr>
<td>No adaptation</td>
<td>27.4%</td>
<td>–</td>
</tr>
<tr>
<td>LFV Modulation</td>
<td>26.3%</td>
<td>–</td>
</tr>
<tr>
<td>Phonetic pre-training</td>
<td>25.4%</td>
<td>–</td>
</tr>
<tr>
<td>Network Superstructure</td>
<td>24.2%</td>
<td>23.5%</td>
</tr>
</tbody>
</table>

- LM1: Baseline
- LM2: Optimized number of BiLSTM cells
Conclusion

- Language adaptation of neural networks
  - Language codes extracted by ancillary network
- Modulation stimulates neural networks to learn features depending on language properties
- Network superstructure with pre-trained sub nets
  - Joint optimization for best recognition performance
- Modulation enables mode dependent networks
  - Intelligent “dropout”
  - Apply method to other domains
- Use more languages: better generalization across languages
Thank you!

## Results

- Network superstructure and NLCs improve performance
  - Evaluation on English

<table>
<thead>
<tr>
<th>Setup</th>
<th>WER LM1</th>
<th>WER LM2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monolingual baseline</td>
<td>25.3%</td>
<td>24.2%</td>
</tr>
<tr>
<td>No adaptation</td>
<td>27.4%</td>
<td>–</td>
</tr>
<tr>
<td>LFV Modulation</td>
<td>26.3%</td>
<td>–</td>
</tr>
<tr>
<td>Phonetic pre-training</td>
<td>25.4%</td>
<td>–</td>
</tr>
<tr>
<td><strong>Network Superstructure</strong></td>
<td><strong>24.2%</strong></td>
<td><strong>23.5%</strong></td>
</tr>
</tbody>
</table>

- **LM1**: Baseline
- **LM2**: Optimized number of BiLSTM cells
Network Architecture

- Stack outputs of subnets
  - Language dependent
  - Remove output layers
  - Stack outputs of last hidden layers

- Main network
  - 2 BiLSTM blocks

- Joint training of all networks
  - Update pre-trained language dependent networks
  - Update NLCs
Results Phonetic Pre-training

- Pre-training the net lowers CER
  - Contrastive experiment: more layers

<table>
<thead>
<tr>
<th>Setup</th>
<th>DE</th>
<th>EN</th>
<th>FR</th>
<th>TR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (4L)</td>
<td>7.7%</td>
<td>11.0%</td>
<td>8.8%</td>
<td>6.0%</td>
</tr>
<tr>
<td>Baseline (6L)</td>
<td>9.0%</td>
<td>12.7%</td>
<td>10.3%</td>
<td>7.5%</td>
</tr>
<tr>
<td>Pre-training (6L)</td>
<td>6.0%</td>
<td>10.0%</td>
<td>8.7%</td>
<td>5.3%</td>
</tr>
</tbody>
</table>

- Evaluation on English, WER

<table>
<thead>
<tr>
<th>Setup</th>
<th>EN CER</th>
<th>EN WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>11.0%</td>
<td>26.3%</td>
</tr>
<tr>
<td>Pre-training</td>
<td>10.0%</td>
<td>25.4%</td>
</tr>
</tbody>
</table>
Results

- Network superstructure and NLCs improve performance
  - Evaluation on English

<table>
<thead>
<tr>
<th>Setup</th>
<th>WER LM1</th>
<th>WER LM2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monolingual baseline</td>
<td>25.3%</td>
<td>24.2%</td>
</tr>
<tr>
<td>No adaptation</td>
<td>27.4%</td>
<td>-</td>
</tr>
<tr>
<td>LFV Modulation</td>
<td>26.3%</td>
<td>-</td>
</tr>
<tr>
<td>Phonetic pre-training</td>
<td>25.4%</td>
<td>-</td>
</tr>
<tr>
<td><strong>Network Superstructure</strong></td>
<td><strong>24.2%</strong></td>
<td><strong>23.5%</strong></td>
</tr>
</tbody>
</table>

- LM1: Baseline
- LM2: Optimized number of BiLSTM cells
Comparison Additive and Multiplicative Codes

- Multilingual systems, Character Error Rate (CER)
- Applying language codes deeper into the network improves performance

<table>
<thead>
<tr>
<th>Adaptation</th>
<th>⌀</th>
<th>ML</th>
<th>DE</th>
<th>EN</th>
<th>FR</th>
<th>TR</th>
</tr>
</thead>
<tbody>
<tr>
<td>No adaptation</td>
<td>14.4%</td>
<td>10.6%</td>
<td>18.2%</td>
<td>15.9%</td>
<td>9.1%</td>
<td></td>
</tr>
<tr>
<td>Additive</td>
<td>13.0%</td>
<td>9.5%</td>
<td>16.1%</td>
<td>14.3%</td>
<td>8.1%</td>
<td></td>
</tr>
<tr>
<td>Multiplicative</td>
<td>12.4%</td>
<td>9.1%</td>
<td>15.5%</td>
<td>13.6%</td>
<td>8.0%</td>
<td></td>
</tr>
</tbody>
</table>