Multilingual Speech Recognition With A Single End-To-End Model

Shubham Toshniwal¹, Tara N. Sainath², Ron J. Weiss², Bo Li², Pedro Moreno², Eugene Weinstein², and Kanishka Rao²

¹TTI Chicago
²Google

April 18, 2017
Why Multilingual Speech Recognition Models?

- Remarkable progress in speech recognition in past few years
- Most of this success restricted to high resource languages, e.g. English
- Google Voice Search supports \( \sim 120 \) out of 7000 languages
- Multilingual models:
  - Utilize knowledge transfer across languages, and thus *alleviate data requirement*
  - Successful in Neural Machine Translation (Google NMT)
  - Easier to deploy and maintain
Conventional ASR Systems

- Traditional ASR systems are modular
- Require expert curated resources
Conventional ASR Systems

- Traditional ASR systems are modular
- Require expert curated resources

Multilingual models:
- Focus on just the acoustic model (Lin, 2009; Ghoshal, 2013)
- Separate language model and pronunciation model required for each language
End-to-end ASR Models

- Encoder-decoder models achieved state-of-the-art result on Google Voice Search task (Chiu et al. 2018)
- Encoder-Decoder models are appealing because:
  - Conceptually simple; subsume the acoustic model, pronunciation model, and language model in a single model.
  - No need for expert curated resources!
We use attention-based encoder-decoder models

Decoder outputs one character per time step

For multilingual models, take union over character sets
Multilingual Encoder-Decoder Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Training</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint model</td>
<td>No language ID</td>
<td>No language ID</td>
</tr>
</tbody>
</table>

- **Naive model**: unaware of multilingual nature of data
- **Can potentially handle code-switching**
## Multilingual Encoder-Decoder Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Training</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint model</td>
<td>No language ID</td>
<td>No language ID</td>
</tr>
<tr>
<td>Multitask model</td>
<td>Language ID</td>
<td>No language ID</td>
</tr>
</tbody>
</table>

- Trained to jointly recognize language ID and speech
## Multilingual Encoder-Decoder Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Training</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint model</td>
<td>No language ID</td>
<td>No language ID</td>
</tr>
<tr>
<td>Multitask model</td>
<td>Language ID</td>
<td>No language ID</td>
</tr>
<tr>
<td>Conditioned model</td>
<td>Language ID</td>
<td>Language ID</td>
</tr>
</tbody>
</table>

- Learnt embedding of language ID fed as input to condition the model
- Language ID embedding can be fed in: (a) Encoder, (b) Decoder, (c) Encoder & Decoder
Encoder-Conditioned Model

Encoder of encoder-conditioned model
Task

- Recognize 9 Indian languages with a single model

<table>
<thead>
<tr>
<th>Language</th>
<th>Script</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bengali</td>
<td>আমার বাবা ওদেরকে বলতেন</td>
</tr>
<tr>
<td>Gujarati</td>
<td>હું ધરાવી અંદર ન મળ્યા અને બહાર પડશા ન મળશા</td>
</tr>
<tr>
<td>Hindi</td>
<td>पहले वीडियोग्राफी होगी</td>
</tr>
<tr>
<td>Kannada</td>
<td>ನಮಗೆ ನಮೂನೆಯ ಎಂಬಿ</td>
</tr>
<tr>
<td>Malayalam</td>
<td>പെഡ്രിഡ് അക്കബുക്ക് സോക്കിന്റെ സഞ്ചാരം ബാലുകളുടെ ശ്രീകൃഷ്ണാച്യ ഗോകുലതല്യ</td>
</tr>
<tr>
<td>Marathi</td>
<td>श्रीकृष्णाच्या गोकुलतल्य</td>
</tr>
<tr>
<td>Tamil</td>
<td>நிற்க நூறு நஞ்சைச்சறு</td>
</tr>
<tr>
<td>Telugu</td>
<td>చెప్పించండి 'పాండా' ప్రమాణంలో శుభేష్ స్థానం చేసండి</td>
</tr>
<tr>
<td>Urdu</td>
<td>شیخ عبدالrahوم گر هوئی چو کلام مصنف</td>
</tr>
</tbody>
</table>

- Very little script overlap, except for Hindi and Marathi.
- The union of character sets is close to 1000 characters!
- But the languages have large overlap in phonetic space (Lavanya et al. 2005).
Experimental Setup

- Training data consists of dictated queries
- Average 230K queries (∼170 hrs) per language

Baseline: Encoder-decoder models trained for individual languages
Joint vs Individual

- Joint model outperforms individual models on all languages!!
- The joint model is not even language aware at test time
- Overall a 21% relative reduction in Word Error Rate (WER)
Picking the Right Script

Rarely confused between languages
Insignificant gains from multitask training
Joint vs Conditioned Models

- As expected, conditioning the model on the language ID of speech helps.
- Encoder conditioning:
  - Performs better than decoder conditioning.
  - Potential acoustic model adaptation happening.
Magic of Conditioning
Testing the Limits: Code Switching

- Can the joint model code switch between 2 Indian languages (trained for recognizing them separately)
Testing the Limits: Code Switching

- Can the joint model code switch between 2 Indian languages (trained for recognizing them separately)?
- Artificial test set of 1000 utterances of Tamil query followed by Hindi with 50ms silence in between
- The model does not code-switch :(
- Picks one of the two scripts and sticks with it
- From manual inspection:
  - Transcribes either the Hindi/Tamil part in corresponding script
  - Transliteration in rare cases
Feeding the Wrong Language ID

▶ Does the model obey acoustics or is it faithful to language ID?
Feeding the Wrong Language ID

- Does the model obey acoustics or is it faithful to language ID?
- Artificial dataset of 1000 Urdu queries tagged as Hindi
- Transliterates Urdu queries in Hindi’s script
- Learns to disentangle the acoustic-phonetic content from the language identity
- Transliterator as a byproduct!
Conclusion

- Encoder-Decoder models:
  - Elegant and simple framework for multilingual models
  - Outperform models trained for specific languages
  - Rarely confused between individual languages
  - Fail at code-switching

- Recent work along similar lines got promising results as well (Watanabe, 2017; Kim, 2018; Dalmia, 2018; Tong, 2018)

- Questions?
Conditioning decoder on top of conditioning the encoder doesn’t buy us much

Possibly because the attention mechanism feeds in information from the encoder to the decoder