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

Shubham Toshniwal[1], Tara N. Sainath[2], Ron J. Weiss[2], Bo Li[2], Pedro Moreno[2], Eugene Weinstein[2], and Kanishka Rao[2]

[1]TTI Chicago

[2]Google

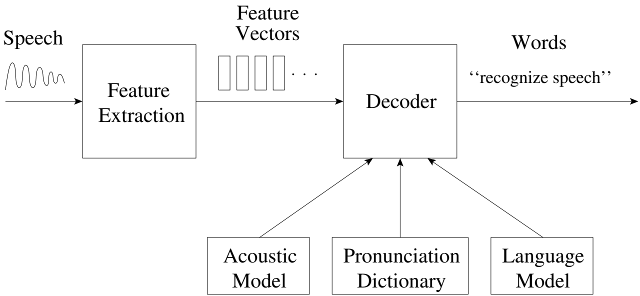
April 18, 2017

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

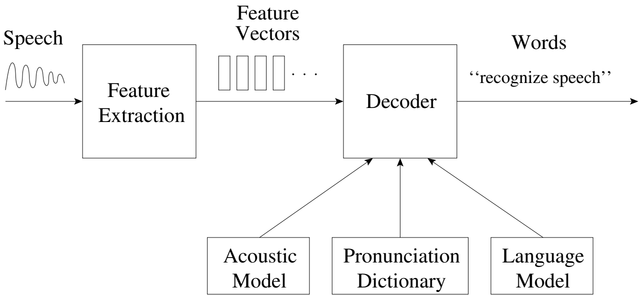
Conventional ASR Systems

- Traditional ASR systems are modular
- Require expert curated resources



Conventional ASR Systems

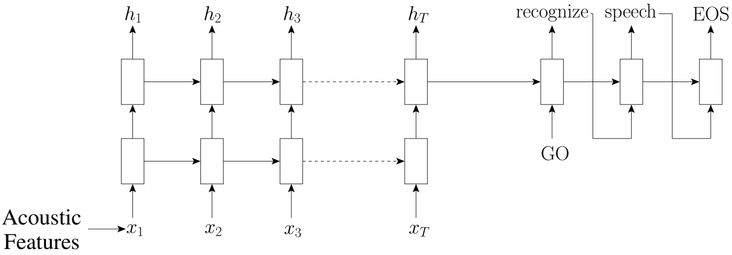
- Traditional ASR systems are modular
- Require expert curated resources



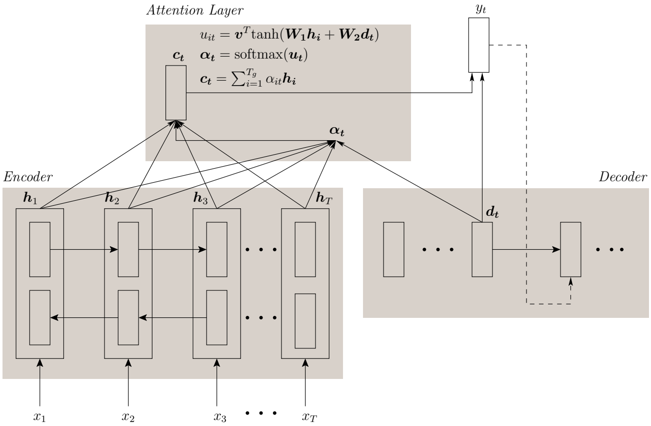
- Multilingual models:
  - Focus on just the acoustic model (Lin, 2009; Ghoshal, 2013)
  - Separate language model and pronunciation model required for each language

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!



End-to-End Multilingual ASR Models



- 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

Model	Training	Inference
Joint model	No language ID	No language ID

- Naive model; unaware of multilingual nature of data
- Can potentially handle code-switching

Multilingual Encoder-Decoder Models

Model	Training	Inference
Joint model	No language ID	No language ID
Multitask model	Language ID	No language ID

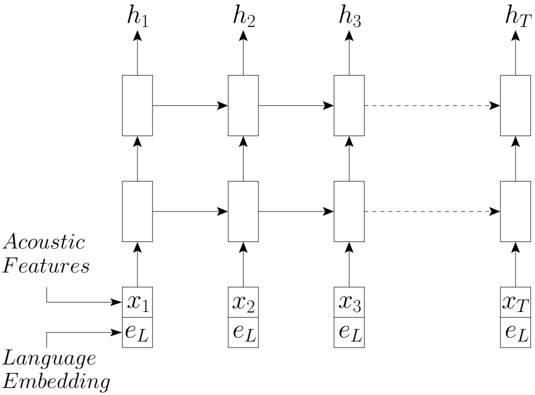
- Trained to jointly recognize language ID and speech

Multilingual Encoder-Decoder Models

Model	Training	Inference
Joint model	No language ID	No language ID
Multitask model	Language ID	No language ID
Conditioned model	Language ID	Language ID

- 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

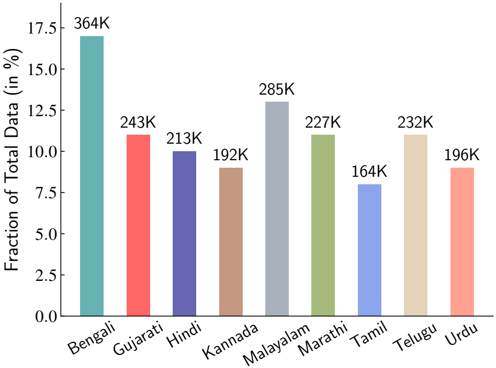
- Recognize 9 Indian languages with a single model

Bengali	আমার বাবা ওদেরকে বলতেন
Gujarati	હું ઘરની અંદર ન મરું અને બહાર પણ ન મરું
Hindi	पहले वीडियोग्राफी होगी
Kannada	ಮುಖದ ಮಧ್ಯದಲ್ಲಿ ಪಿಷ್ಟ
Malayalam	എന്നിട്ടും അവരുടെ വാക്കുകളിലൂടെ അവരെ അറിയുന്നുണ്ട്
Marathi	श्रीकृष्णाच्या गोकुळातल्या
Tamil	இது ஒரு நகராட்சியாகும்
Telugu	ఈ పేజీని 'తర్జుమా' చేయకముందు ఇవికీలో పెడదామా
Urdu	شیخ عبدالرحیم گرھوڑی جو کلام مصنفہ

- 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).

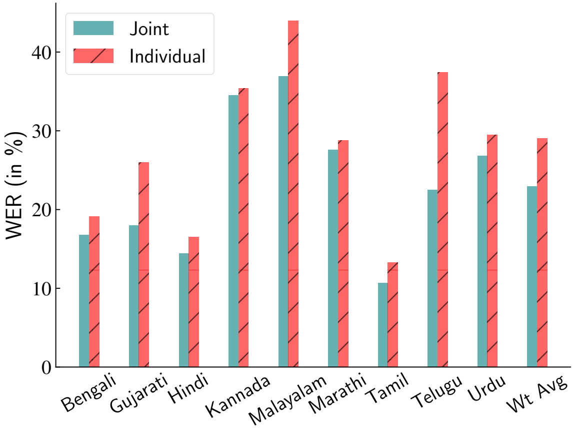
Experimental Setup

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



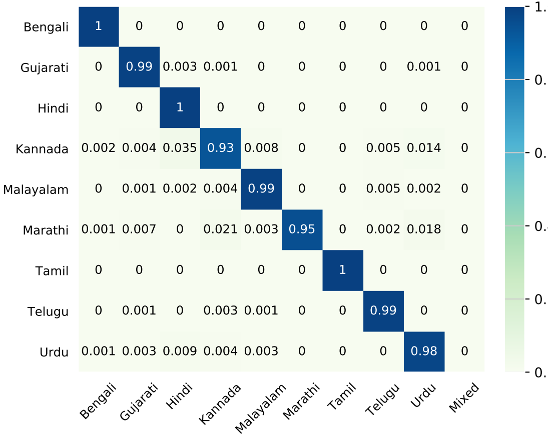
- Baseline: Encoder-decoder models trained for individual languages

Joint vs Individual



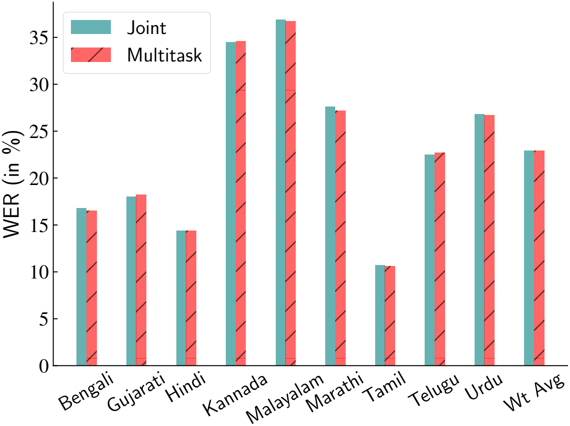
- Joint model outperforms individual models on all languages!!
- The joint model is not even language aware at test time
- Overall a 21% relative reduction in Word Error Rate (WER)

Picking the Right Script



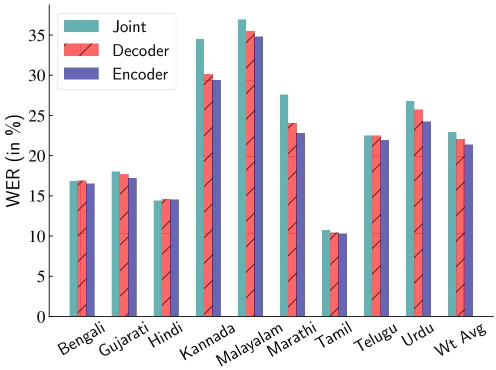
Rarely confused between languages

Joint vs Multitask



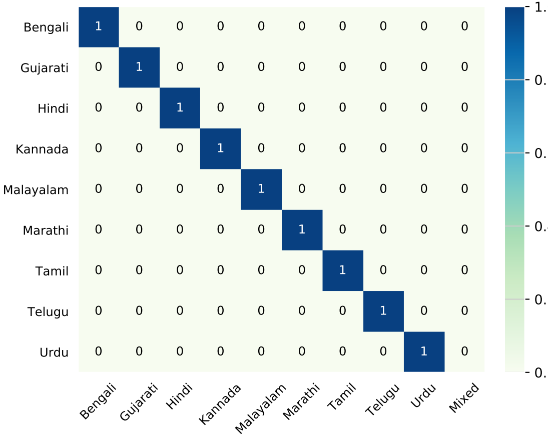
Insignificant gains from multitask training

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?

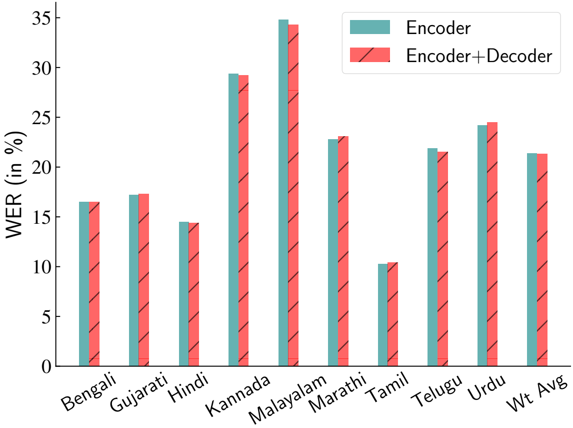
Feeding the Wrong Language ID

- 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 Encoder is Enough



- 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