Incorporating AI-based Speech Transcription into Language Documentation

A case study of Imbabura Kichwa

Chihiro Taguchi (田口智大)
Department of Computer Science and Engineering, University of Notre Dame

Abstract

Problem:
- Half of the world's languages might be extinct by the next century[1]
- Transcription process is the biggest bottleneck in language documentation
  - ~2weeks to transcribe 1-hour audio[2]
- Lack of field linguists
- Lack of funding

Solution:
- **Transcription with the state-of-the-art automatic speech recognition (ASR) model** (like Siri, Alexa, or YouTube's automatic subtitles)
- This study shows a successful case study of building an ASR model for transcribing the Kichwa language with limited audio resources

Introduction

Imbabura Kichwa:
- < Kichwa < Northern Quechua (Quechua II-B) < Quechua II < Quechua
- Spoken in the Imbabura Province of Ecuador
- Spoken by ~150,000 speakers[3] (debatable; probably underestimated)
- Socially stigmatized; ongoing language shift to Spanish
- Few linguistic research works despite the size of the speaker community

Phonology and orthography
- Imbabura Kichwa phonology is relatively simple
  - 16~20 consonants, 3 vowels[3]
  - CV(C)
- Unified orthography for Ecuadorian Kichwa

Data
- No publicly available ASR model for Kichwa
- No ASR dataset

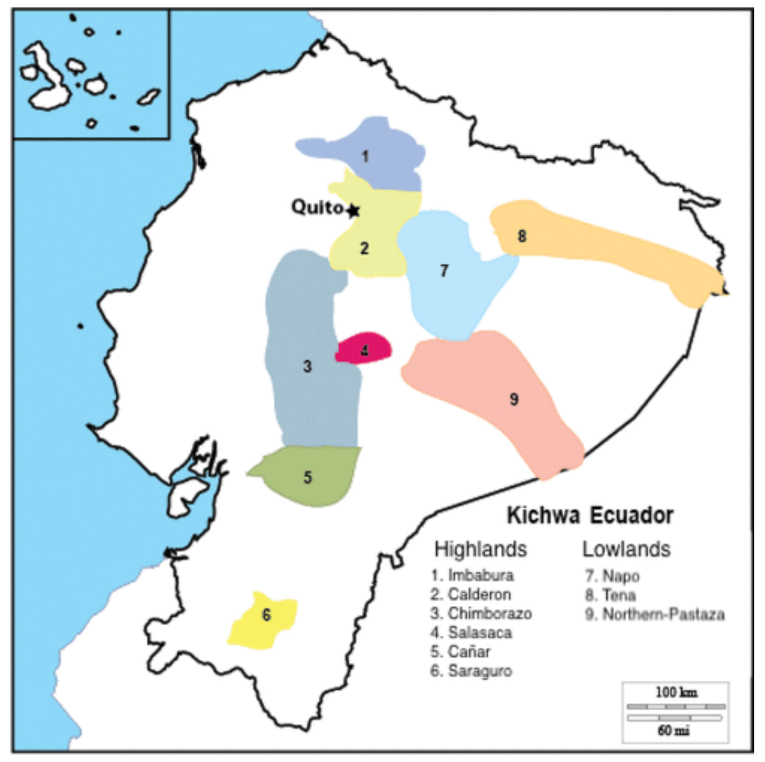

Figure 1. Distribution of Ecuadorian Kichwa varieties[3]

Methods

Objective: Fine-tuning the multilingual ASR model (Figure 2)
- Pre-trained model: Wav2vec2-large-xlsr-53, developed by Meta AI[4]
  - Trained on 53 languages
  - Able to represent multilingual speech
  - Connectionist Temporal Classification (CTC)
- Create the Kichwa dataset
  - Radio program in Kichwa (Creative Commons BY-SA)[5]
  - Add the transcription with **ELAN**
  - Python code to trim and save each audio–annotation segment[6]
- Train the model with 1–4 episodes (~14 min. per episode)[7]
  - With different epochs (epoch: number of training cycles)
- Trained on 2 GPUs (Quadro RTX 6000), max. ~63 mins.

Evaluation:
- The 5th episode is reserved for the test dataset
- The accuracy metric is Character Error Rate (CER)[8]
  - $CER = \frac{(S+D+I)}{N}$ (S: substitutions, D: deletions, I: insertions, N: reference string length)
  - E.g., "language" vs. "linguam": S=1, D=2, I=1, N=8 → CER = 4/8 = 0.5

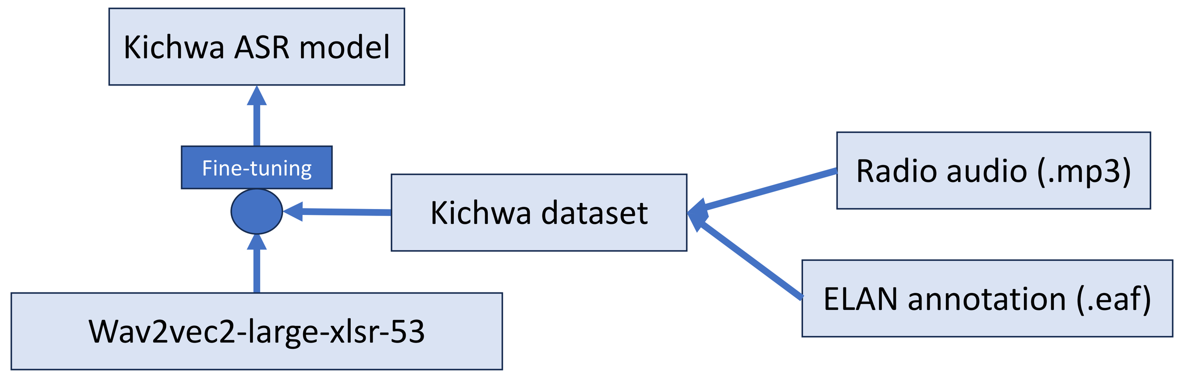

Figure 2. Illustration of the workflow.

Results

See Table 1.
- Best score: 4 episodes, 30 epochs (~**92% correct output**)[9]
  - **Less than 1 hour** of training data!
- The **more data** we have, the **better accuracy** we get
- Too many epochs can harm the accuracy (overfitting)

485 sec. for transcribing 820 sec. test data

Table 1. Comparison of Character Error Rates with different dataset sizes and epochs. The unit is %.

	1 episode	2 episodes	3 episodes	4 episodes
20 epochs	—	15.24	11.44	10.04
30 epochs	—	12.11	11.56	**8.16**
40 epochs	18.29	13.65	10.24	8.29

Discussion

From the Results:
- We can develop a good ASR system with 1-hour training data!
- A possible workflow:
  1. Manual annotation (1 hour)
  2. Train an ASR model
  3. Get a draft transcription
  4. Post-edit
  5. Re-train the model (repeat from 3.)

1-hour audio file would be transcribed in ~35.5 minutes
- **EASY: Annotators can sip a cup of coffee** during the process!
- **FAST: Drastic acceleration** compared to 2 weeks of manual transcription
- **CHEAP:** Audio doesn't have to be of high quality
  - Speakers can record their speech with their own device (e.g., via Whatsapp voice message; exportable to .wav)
  - Flexibility for **remote fieldwork**

Discussions for future work
- How about more phonologically/orthographically complex languages?
- How about tonal languages?
- Can such a model be used for code-mixed speech?
- The ownership of the audio data must be carefully discussed with informants.
- How can this technology contribute to the local speaker community?

Limitations
- Some coding is necessary
- Model size is huge (~1.2GB)
- Access to GPUs is necessary (expensive!)
  - These can be overcome by collaborations

Concluding remarks

This study showed…
- A successful case study of developing a Kichwa ASR model for language documentation
  - Only 1-hour audio is necessary to achieve >90% accuracy
  - Feasible workload for field linguists
- Contribution to applications in Kichwa, an underrepresented language in technology
- Bridging between field linguistics and natural language processing (NLP)

Takeaways:
- NLP has great potential for language documentation!
- Call for collaborative works of linguistics and NLP

Questions/Collaborations? Contact me!

Chihiro Taguchi
University of Notre Dame
Address: Fitzpatrick Hall of Engineering, Notre Dame, IN 46556
Email: ctaguchi@nd.edu
Website: https://ctaguchi.github.io

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. BCS-2109709.
The fieldwork in Ecuador was supported by the Summer Language Abroad Grant provided by the Center for the Study of Languages and Cultures (CSLC) of the University of Notre Dame.
Also, the Kellogg Institute for International Studies partially supported my participation in the LSA Institute.
I am grateful for these supporters, my Kichwa teachers, Jefferson and Ángel, and my host family for my fieldwork in Quito.

References

1. Austin, P. K. and McGill, S. (2011). Endangered Languages. Routledge.
2. Clifton Pye and Pedro Mateo Pedro, p.c. at the LSA Institute.
3. Gualapuro Gualapuro, S. D. (2017). Imbabura Kichwa phology. Master's thesis, University of Texas at Austin.
4. Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. CoRR, abs/2006.11477.
5. https://radialistas.net/jaboneropak-ayllullaktapi/
6. Conversion code is available here: https://gist.github.com/ctaguchi
7. Training code is available here: https://github.com/ctaguchi/kichwaasr
8. https://huggingface.co/spaces/evaluate-metric/cer
9. The best model is available here: https://huggingface.co/ctaguchi/kichwaasr5