User-Friendly Automatic Transcription of Low-Resource Languages: Plugging ESPnet into Elpis

Oliver Adams, Benjamin Galliot, Guillaume Wisniewski, Nicholas Lambourne, Ben Foley, Rahasya Sanders-Dwyer, Janet Wiles, Alexis Michaud, Séverine Guillaume, Laurent Besacier, Christopher Cox, Katya Aplonova, Guillaume Jacques, Nathan Hill

French National Centre for Scientific Research (CNRS),
University of Queensland,
School of Oriental and African Studies (SOAS) University of London,
Center of Excellence for the Dynamics of Language (CoEDL),
Atos zData,
University of Alberta

Advances in speech recognition models

- Performance improvements for high and low-resource languages.
  - Speech recognition on big languages is now an everyday reality for a lot of people.
  - Advancements in low-resource models too.
- End-to-end neural models have featured prominently in recent research
  - No lexicon required
  - Burgeoning research in self-supervised pre-training on speech.
- Numerous open-source tools for end-to-end ASR.
  - wav2letter, deep speech, espnet.

But not much in the way of useable tools

- Some more user-friendly options:
  - www.dictate.app for narrow phonetic transcription online with a pre-existing model.
  - Persephone-ELAN as an ELAN plug-in for Persephone.
  - Elpis as a user-friendly web front-end for training Kaldi models.
- Wouldn't it be nice if Elpis had access to a broader range of state of the art models, some that don't require a lexicon?

Backstory on the importance of a good UI

- During my PhD I did some phonemic transcription experiments.
- Code was packaged up as Persephone for others to use.
- However, usability remained an issue.
- Being involved in an in-person workshop highlighted this to me:
  - Issues with installation
  - Issues because of required Python knowledge.
  - Issues to do with other knowledge assumed by documentation.
  - ...

So what do we want?

A speech recognition tool with:

- A UI for training/transcription that is as simple as possible to navigate (Elpis)
- State of the art end-to-end neural models (ESPnet)
- Potential for hosting on a server so users don't need to deal with installation

In the rest of this talk I discuss our progress in incorporating ESPnet into Elpis, and some other changes we made to Elpis along the way.

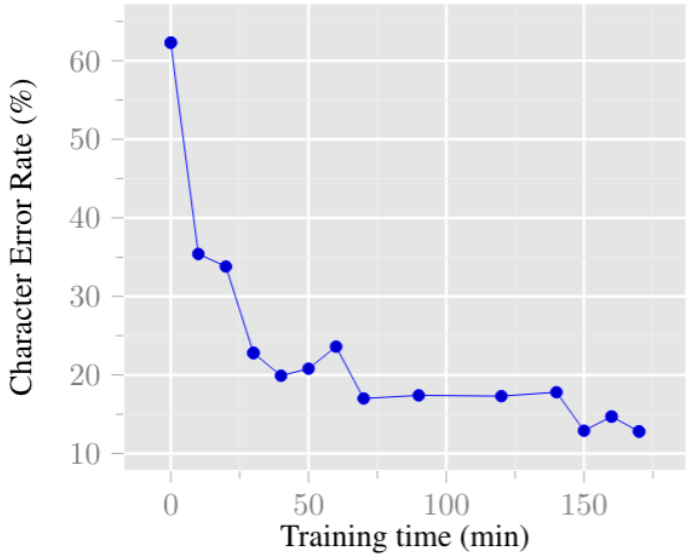
Developing an ESPnet recipe

- We created an ESPnet recipe to loosely reproduce transcription accuracy of Persephone models while instead using a more efficient architecture.
- This recipe is on standby for future refinement.
- We moved to evaluating in terms of character error rate.

Language	Num speakers	Type	Train (minutes)	CER (%)
Na	1	Spontaneous narratives	273	14.5
Na	1	Elicited words & phrases	188	4.7
Chatino	1	Read speech	81	23.5
Japhug	1	Spontaneous narratives	170	12.8

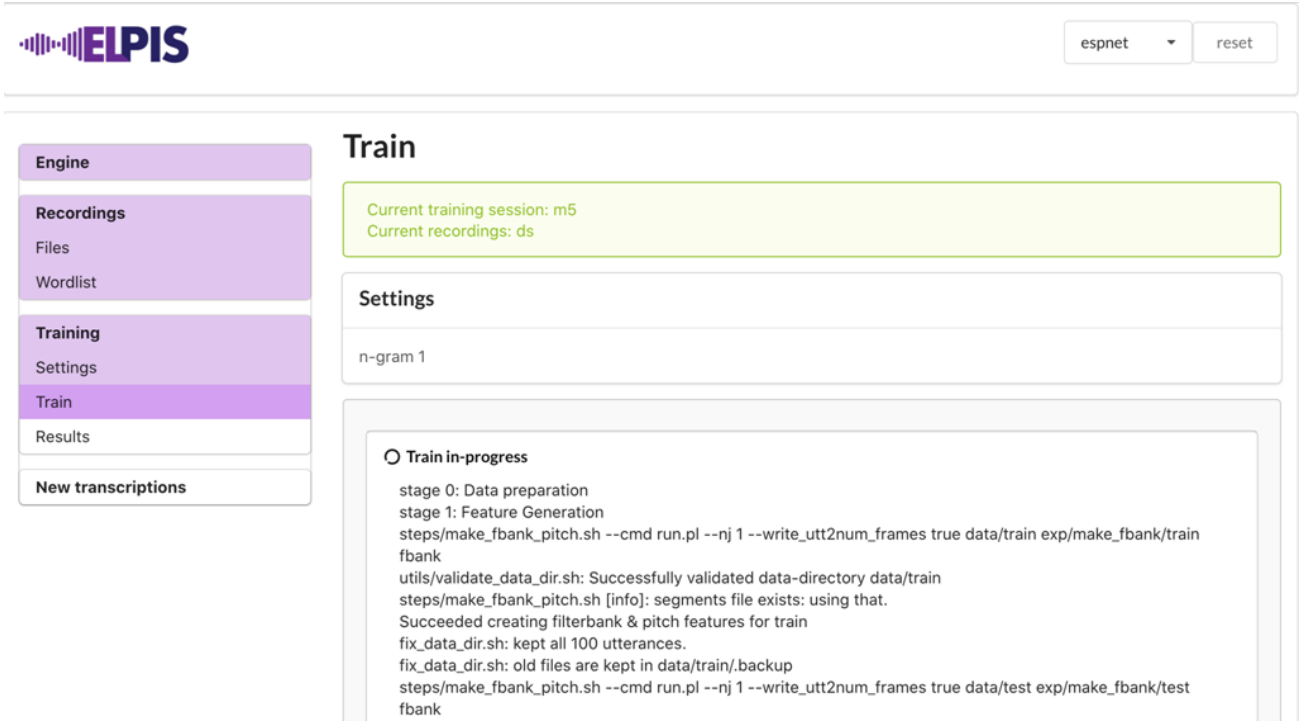
Japhug experiment

- A toneless Sino-Tibetan language
- A rich system of consonant clusters
- 'Flamboyant' morphology :)



Plugging Elpis into this ESPnet recipe

We then went about connecting the front-end to that backend.



Other Elpis enhancements

Other enhancements:

- CUDA-supported docker image to leverage GPUs.
- i18n. French language UI and it is now easier to add support for other languages.

Going forward

Going forward:

- Iterative improvement of:
  - The underlying ESPnet recipe
  - The Elpis user interface
- Exploring incorporation of self-supervised pre-training as a means of making the tool:
  - Perform better with limited transcribed audio training data
  - Take advantage of untranscribed audio in a target language
- Elpis as a web service.

Try it out!

Elpis documentation is at: https://elpis.readthedocs.io/en/latest/

The Elpis github page is at: https://github.com/CoEDL/elpis

Reach out to us with your experiences: b.foley@uq.edu.au

There will also be an Elpis workshop at the International Conference on Language Documentation and Conservation (ICLDC) 4-7th March.