The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages
29-31 August 2018, Gurugram, India

Building Speech Recognition Systems for Language Documentation: The CoEDL Endangered Language Pipeline and Inference System (Elpis)

Ben Foley[19], *Josh Arnold*[19], *Rolando Coto-Solano*[29], *Gautier Durantin*[19], *T. Mark Ellison*[39], *Daan van Esch*[4], *Scott Heath*[19], *František Kratochvíl*[5], *Zara Maxwell-Smith*[39], *David Nash*[39], *Ola Olsson*[19], *Mark Richards*[69], *Nay San*[39], *Hywel Stoakes*[789], *Nick Thieberger*[79], *Janet Wiles*[19]

[1] The University of Queensland, Australia
[2] Victoria University of Wellington, New Zealand
[3] Australian National University, Australia
[4] Google, Mountain View (CA), USA
[5] Palacký University, Czech Republic
[6] Western Sydney University, Australia
[7] The University of Melbourne, Australia
[8] The University of Auckland, New Zealand
[9] ARC Centre of Excellence for the Dynamics of Language (CoEDL), Australia

b.foley@uq.edu.au, j.arnold4@uq.edu.au, rolando.coto@vuw.ac.nz, g.durantin@uq.edu.au, m.ellison@anu.edu.au, dvanesch@google.com, scott.heath@uq.edu.au, frantisek.kratochvil@upol.cz, zara.maxwell-smith@anu.edu.au, david.nash@anu.edu.au, o.olsson@uq.edu.au, m.richards@westernsydney.edu.au, nay.san@anu.edu.au, h.stoakes@auckland.ac.nz, thien@unimelb.edu.au, j.wiles@uq.edu.au

Abstract

Machine learning has revolutionised speech technologies for major world languages, but these technologies have generally not been available for the roughly 4,000 languages with populations of fewer than 10,000 speakers. This paper describes the development of Elpis, a pipeline which language documentation workers with minimal computational experience can use to build their own speech recognition models, resulting in models being built for 16 languages from the Asia-Pacific region. Elpis puts machine learning speech technologies within reach of people working with languages with scarce data, in a scalable way. This is impactful since it enables language communities to cross the digital divide, and speeds up language documentation. Complete automation of the process is not feasible for languages with small quantities of data and potentially large vocabularies. Hence our goal is not full automation, but rather to make a practical and effective workflow that integrates machine learning technologies.

Index Terms: speech recognition, language documentation, low-resource languages, linguistic fieldwork, machine learning

1. Introduction

While approximately 100 languages globally have access to commercial speech technologies such as Google's speech-recognition systems [1], the benefits of making speech tools accessible for more languages would be significant for language speakers and the broader language-research community. Extending the coverage of automatic speech recognition (ASR) systems to languages with speaker-bases which have traditionally been too small to access these technologies can help assist their first-language communication on modern devices like smartphones, boosting language status and contributing to ongoing language use. In addition, access to ASR tools for more languages would enable the use of much bigger data sets in language documentation by speeding up transcription, radically increasing transcription productivity throughput by pre-filling with results of machine learning processes [2].

In this paper, we focus on building ASR systems for the language documentation use case, although the systems we built should also be usable by language communities at large. We will describe the context for this work, with a brief introduction to the challenge of transcription for language documentation. Then, we will describe how we implemented a pipeline to accelerate the transcription process, and show ASR results across a range of low-resource languages for which no ASR systems have ever been trained. Our pipeline can easily be used to transcribe fieldwork recordings from any language, with ingestion support for all major fieldwork transcription tools, and is accessible to linguists without a machine learning background.

2. Background

2.1. Transcription bottleneck

Fieldworkers working on language documentation projects frequently collect hundreds of hours of speech recordings, and often work on low-resource languages:

10.21437/SLTU.2018-42

they may even work on languages that have never been recorded before [3] [4] [5]. The process of transcribing this speech data collected in the field is laborious. In a 2017 survey of 51 linguists, Durantin [6] found that one minute of data takes 40 minutes to transcribe on average, with a long tail of lengthier times for difficult transcriptions. As a result, manual transcription is a major bottleneck for language workers wanting to work with the written form of these low-resource languages, either for new language documentation, or when working through archives of existing recordings. This bottleneck prevents many linguists from working with more than a small fraction of their corpora each year. In addition, the slow process of manual transcription strangles the use of existing collections of language material in archives such as PARADISEC [7] and AIATSIS [8], containing approximately 8,300 and 40,000 hours of speech data respectively, from being used in language resource production, e.g. the creation of dictionaries or reference grammars.

2.2. Existing technologies

Machine Learning (ML) technologies are available for some languages through commercial ASR systems. For example, Google currently provides a cloud system for 119 language varieties through its Cloud Speech-to-Text API. For languages already serviced by Google, Amazon, IBM, Nuance, Microsoft or other commercial providers, these are viable technologies which language workers can readily use for speeding up transcription, with minimal technical knowledge required. In addition, open-source technologies are available to train models for individual languages, using technologies such as the Kaldi speech recognition toolkit [9]. However, implementing or using these systems requires significant technical experience. Our work focussed on bringing open-source ASR technologies within reach of language workers who may not have the technical experience required to either set up or operate bespoke systems.

3. Our solution: the Elpis pipeline

To make it easy for language documentation workers to build turn-key models for speech recognition in any language they work on, we have built Elpis (the Endangered Language Pipeline and Inference System), a pipeline of tools which enable people with minimal technical knowledge to train ASR models on an existing transcribed subset from a set of fieldwork recordings, and to apply the resulting system to obtain a first-pass transcription on untranscribed recordings.

This is, to our knowledge, the first project to explore the use of ASR systems for endangered language documentation across a wide range of languages. We believe that one of the main reasons this type of work had not been done before is that unfortunately, it remains relatively rare for language documentation fieldworkers, computational linguists, software engineers and machine learning researchers to interact. We organised a number of workshops bringing these groups together.

4. Our first workshop: Start of technical development

In the first workshop, hosted by the ARC Centre of Excellence for the Dynamics of Language (CoEDL), we started developing the Elpis pipeline. Participants worked in four groups, with differing goals:

- Set up the Kaldi open-source ASR system
- Organise and select appropriate data from corpora of fieldwork recordings
- Clean and normalise the selected input data
- Develop pronunciation rules for Kaldi to use in training the models

4.1. Data standardisation

One key challenge we encountered in this workshop was data standardisation. Language documentation workers use various types of recording devices and store their corpora in a variety of formats. We built a pipeline for standardizing these recordings, which resamples audio files to a consistent sample rate, bit rate and codec (44.1kHz, 16bit, WAV). In addition, our ingestion pipeline allows the conversion of any existing transcriptions from the three input formats typically used in a linguist's workflow (ELAN [10] [11], PRAAT [12], Transcriber [13]) into a JSON interchange format. The transcriptions that linguists create from field recordings are commonly time-aligned, allowing for easy separation of long audio files into their component utterances. After conversion, the interchange format contains all the content which Kaldi expects, and even includes other data from the input files that are not relevant to Kaldi, making this standardised format useful and usable in other language resource production workflows.

4.2. Data normalisation

Once the audio has been converted and the standardised JSON file has been created with time-aligned transcriptions, normalisation scripts process the text for language model training. Our system removes punctuation, non-orthographic word forms (e.g. comments marked with "#"), and English words (using the NLTK toolkit [14]), while also lowercasing the text. Further scripts derive a vocabulary list and create the pronunciation lexicon based on a supplied set of manually specified grapheme-to-phoneme (G2P) mappings.

Our set-up currently uses a single normalisation module for stripping out non-linguistic content from the transcriptions, while the G2P mappings were created individually by each linguist for their own language data. As the languages we worked on typically had transparent orthographies, this was a relatively straightforward task. We did not handle verbalization of numbers and other not-a-word tokens since these rarely occurred in the data. Where they did, our tool skipped the utterance.

4.3. Train/test split

Finally, we also applied a test-train split with 90% of audio recordings and transcriptions assigned to the train set, while 10% was held out for testing. We should note that there is typically some overlap in vocabulary

and even in speakers. For our purposes, this is acceptable, as it reflects the target use case: the audio corpora are typically limited to a relatively small speaker population, as language documentation fieldworkers usually work closely with a small group of speakers, and the goal is chiefly to transcribe the untranscribed fieldwork recordings, which also include the same speakers observed in the train set. Once this split has been created, a script prepares the Kaldi workspace by moving files into the locations which Kaldi expects. By the end of this workshop, preliminary results were obtained for Abui (ISO 639: abz; about 17,000 speakers in Indonesia) and Komnzo (no ISO 639 code assigned; about 200 speakers in Papua New Guinea).

5. Containerisation

Following the first workshop, Nay San migrated the pipeline to Docker, a software containerisation platform [15], with the intention of reducing the complexity of setting up Kaldi. Having the pipeline available in a Docker image reduced the install time of Kaldi from the two days experienced in the first workshop down to half an hour in the second. 'Go-task' task runners were used as a simple way to invoke collections of scripts. For example, the manual way of cleaning data for Kaldi requires the language worker to batch convert audio files; check and clean individual transcription files; construct a list of unique words in the corpus; build a pronunciation model representing every word in the corpus and then move files into the correct folders. The potential for human error in this workflow is very high. Instead, these steps, automated by Python scripts, can be grouped by a single task runner command, and invoked by a single command 'task run-elan'.

6. Our second workshop: Scaling to more languages

During the 2018 workshop, engineers and linguists, with a range of computational experience, ran the Elpis system on their own language data. Following a morning of introductory talks relating to ASR and ML tools including the closely related Persephone system [16], the group participated in a workshop on data management and basic methods of running scripts. After practising with a dummy corpus to gain some experience operating the pipeline, the linguists then processed their own data.

During the following week, some of the linguists tuned their system's parameters to try and reduce error rates, and by the time we hosted a follow-up session at the end of the week, linguists had trained language models for 12 languages, with varying error rates. We attribute this variation to a number of factors:

- Differences in quality and quantity of audio data
- Differences in the nature of the fieldwork recordings, e.g. single-word elicitation vs. sentences (longer units, e.g. stories, were broken down into sentences using time alignments)
- Differences in the degree of manual tuning which was done on individual languages (e.g. in the pronunciation model)
- Morphological complexity and or orthographic conventions of some of the target languages, where a bigger inflected word inventory leads to higher OOV rates (similar to effects observed in [17]). Subword language models would help with this, but our current pipeline does not support training such models yet.

Table 1 shows the Word Error Rates (WER) for some of the languages we worked on using Kaldi.

Although training data sizes are relatively small, the vocabulary for some of these data sets is limited, as is the set of speakers, meaning that some of the WER results our systems achieve are still reasonably decent. Certainly, all transcriptions still require human post-editing, but their presence helps accelerate the transcription process, meaning more training data can be created, after which the system can be retrained in order to achieve quality improvements. In particular, linguists found visualizing the output using lattices quite informative to understand the underlying mechanisms at work, and noticed that even when the top hypothesis was wrong, they could frequently find the right words in the lattice, suggesting it would be possible to make use of lattices in the transcription workflow. Figure 1 shows a sample lattice for the Abui language we worked on.

Linguists continued to train systems for new languages following the workshop, with the total now reaching 16.

7. Next steps

Following these workshops, our work has focussed on making the code more robust and building a simple user interface to the code on a hosted server. In the near future, our plans for further extending the pipeline include:

- Showing the model's lattices for the language worker to 'override' the automatic transcription
- Enabling the generation of visualisations of the language data
- Supporting output of inferences in ELAN format
- Benchmarking the time taken to obtain and edit a best-guess transcription using Elpis against the time required to manually transcribe the same data
- Releasing the code and documentation on GitHub

7.1. Lattices and visualisation

Seeing a graph visualisation of the word lattices gave the linguists at the 2018 workshop great insight into the mechanics of the training process. People were able to see that the words in a decoded transcription were determined from a number of options which they would be able to influence by, say, including more training data or changing the pronunciation model. We propose to include alternate word hypotheses in the generated Elan file along with an indication of the confidence value of each option, to give the end user an intuitive way to understand the model, correct transcription errors, and give a sense of how much to trust the generated transcription. Similarly, visualising statistics about the language data (e.g. the number of utterances, the signal-to-noise ratio in the audio, and more) would help provide insight into model behaviour and accuracy.

Language	ISO 639	Processed by	LM n-gram order	Training data	WER
Abui	abz	Durantin	3	2 hours	13.3*
Arrernte	aer	Stoakes	1	< 1 hour	73.4
Bininj Kunwok	gup	Ellison, Marley	1	17 min (narrative, single speaker)	48.1
Bininj Kunwok	gup	Stoakes	1	< 1 hour (word list, 3 speakers)	80.5
Cook Islands Maori	rar	Coto-Solano	3	< 1 hour	37.1
Ende	kit	Ellison	1	11 min	52.9
Indonesian	ind	Maxwell-Smith	3	< 1 hour	48.3
Mangarrayi	mpc	Richards	1	< 1 hour	61.9
Nafsan	erk	Thieberger	3	3 hours	42.7
Warlpiri	wbp	Nash	1	< 1 hour	39.0

Table 1: *Word Error Rates (WERs) for some of the languages we worked on in the second workshop, using Kaldi with a 90/10 train/test split as described above, except for Abui which used a 125/10 train/test split.*

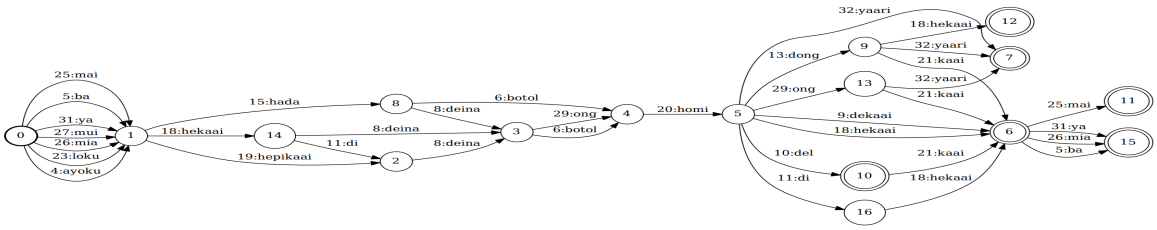

Figure 1: *Lattice for one of the utterances in our Abui corpus.*

7.2. Output into ELAN format

Our pipeline already enables linguists to ingest transcription files from common linguistic tools, making thepipeline highly accessible. Outputting the inference transcriptions to a popular format – time-aligned and exposing the confidence of the transcription – will enable the pipeline to be used in linguists' workflows without the burden of learning yet-another piece of software.

7.3. Longer-term plans

In general, we found that the majority of linguists in the workshop groups were able to operate the scripts with minimal training. However, some participants found system path structures difficult to understand, and were unable to use the pipeline. Others had difficulty in getting Docker to install, due to dated operating systems, or lack of administration permissions for their computers. If we could set up a hosted server to run the pipeline, this might resolve these issues, enabling even more people to use the system.

8. Conclusion

Providing ways for users with minimal technical knowledge to access ASR tools makes a big impact on the reach of these systems into communities who would otherwise not have the opportunity to apply cutting-edge tools to their languages. Our current work has seen exciting initial results for 16 languages from the Asia-Pacific region, enabling linguists with some technical knowledge to train models to obtain usable first-pass transcriptions of their data. With further development, the Elpis system should become usable by language workers without any technical knowledge. We hope Elpis will provide new tools for linguists and communities to document, preserve and future-proof many of the world's endangered languages.

9. References

[1] Google Cloud Speech-to-Text API language support. Google. [Online]. Available: https://cloud.google.com/speech-to-text/docs/languages

[2] L. M. Johnson, M. D. Paolo, and A. Bell, "Forced alignment for understudied language varieties: Testing Prosodylab-Aligner with Tongan data." *Language Documentation & Conservation*, vol. 12, pp. 80–123, 2018.

[3] N. Evans, *Dying Words: Endangered Languages and What They Have to Tell Us*. Wiley-Blackwell, 2011.

[4] F. Meakins, J. Green, and M. Turpin, *Understanding Linguistic Fieldwork*. Routledge, 2018.

[5] M. Carew, J. Green, I. Kral, R. Nordlinger, and R. Singer, "Getting in touch: Language and digital inclusion in Australian Indigenous communities," vol. 9, pp. 307–323, 2015.

[6] G. Durantin, B. Foley, N. Evans, and J. Wiles, "Transcription survey," 2017.

[7] Pacific and Regional Archive for Digital Sources in Endangered Cultures (PARADISEC). [Online]. Available: http://www.paradisec.org.au

[8] Australian Institute of Aboriginal and Torres Strait Islander Studies (AIATSIS). [Online]. Available: https://aiatsis.gov.au

[9] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *IEEE 2011*

Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[10] T. L. A. Max Planck Institute for Psycholinguistics. ELAN. Nijmegen, The Netherlands. [Online]. Available: https://tla.mpi.nl/tools/tla-tools/elan/

[11] H. Brugman and A. Russel, "Annotating multi-media/multi-modal resources with ELAN," in *LREC*, 2004.

[12] P. Boersma and D. Weenink, "Praat, a system for doing phonetics by computer," vol. 5, pp. 341–345, 01 2001.

[13] TranscriberAG: a tool for segmenting, labeling and transcribing speech. [Online]. Available: http://transag.sourceforge.net

[14] E. Loper and S. Bird, "NLTK: the natural language toolkit," *CoRR*, vol. cs.CL/0205028, 2002. [Online]. Available: http://arxiv.org/abs/cs.CL/0205028

[15] Docker container software. [Online]. Available: http://transag.sourceforge.net

[16] O. Adams, T. Cohn, G. Neubig, H. Cruz, S. Bird, and A. Michaud, "Evaluation phonemic transcription of low-resource tonal languages for language documentation," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds. Paris, France: European Language Resources Association (ELRA), May 2018.

[17] R. Cotterell, S. J. Mielke, J. Eisner, and B. Roark, "Are all languages equally hard to language-model?" in *Proceedings of the 2018 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, June 2018.