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, et al.

Accepted at: Fourth Workshop on the Use of Computational Methods in the Study of Endangered Languages (ComputEL-4), March 2-3, 2021.

User-friendly Automatic Transcription of Low-resource Languages: Plugging ESPnet into Elpis

Oliver Adams [a], Benjamin Galliot [b], Guillaume Wisniewski [c], Nicholas Lambourne [d e], Ben Foley [d e], Rahasya Sanders-Dwyer [d e], Janet Wiles [d e], Alexis Michaud [b], Séverine Guillaume [b], Laurent Besacier [f], Christopher Cox [g], Katya Aplonova [h], Guillaume Jacques [i], Nathan Hill [j]

[a] Atos zData, United States of America
[b] Langues et Civilisations à Tradition Orale (LACITO), CNRS-Sorbonne Nouvelle, France
[c] Laboratoire de Linguistique Formelle (LLF), CNRS-Université Paris-Diderot, France
[d] The University of Queensland, Brisbane, Australia
[e] ARC Centre of Excellence for the Dynamics of Language (CoEDL), Australia
[f] Laboratoire d'Informatique de Grenoble (LIG), CNRS-Université Grenoble Alpes, France
[g] University of Alberta, Canada
[h] Langage, Langues et Civilisation d'Afrique (LLACAN), CNRS-INALCO, France
[i] Centre de Recherches Linguistiques sur l'Asie Orientale (CRLAO), CNRS-EHESS, France
[j] School of Oriental and African Studies, University of London, United Kingdom
oliver.adams@gmail.com, b.g01lyon@gmail.com,
guillaume.wisniewski@u-paris.fr,
{n.lambourne|b.foley|uqrsand5|j.wiles}@uq.edu.au,
{alexis.michaud|severine.guillaume}@cnrs.fr,
laurent.besacier@univ-grenoble-alpes.fr, cox.christopher@gmail.com,
{aplooon|rgyalrongskad}@gmail.com, nh36@soas.ac.uk

Abstract

This paper reports on progress integrating the speech recognition toolkit ESPnet into Elpis, a web front-end originally designed to provide access to the Kaldi automatic speech recognition toolkit. The goal of this work is to make end-to-end speech recognition models available to language workers via a user-friendly graphical interface. Encouraging results are reported on (i) development of an ESPnet recipe for use in Elpis, with preliminary results on data sets previously used for training acoustic models with the Persephone toolkit along with a new data set that had not previously been used in speech recognition, and (ii) incorporating ESPnet into Elpis along with UI enhancements and a CUDA-supported Dockerfile.

1 Introduction

Transcription of speech is an important part of language documentation, and yet speech recognition technology has not been widely harnessed to aid linguists. Despite revolutionary progress in the performance of speech recognition systems in the past decade (Hinton et al., 2012; Hannun et al., 2014; Zeyer et al., 2018; Hadian et al., 2018; Ravanelli et al., 2019; Zhou et al., 2020), including in the application to low-resource languages (Besacier et al., 2014; Blokland et al., 2015; Lim et al., 2018; van Esch et al., 2019; Hjortnaes et al., 2020), these advances are yet to play a common role in language documentation workflows. Speech recognition software often requires effective command line skills and a reasonable understanding of the underlying modeling. People involved in language documentation, language description, and language revitalization projects (this includes, but is not limited to, linguists who carry out fieldwork) seldom have such knowledge. Thus, the tools are largely inaccessible by many people who would benefit from their use.

Elpis[1] is a tool created to allow language workers with minimal computational experience to build their own speech recognition models and automatically transcribe audio (Foley et al., 2018, 2019). Elpis uses the Kaldi[2] automatic speech recognition (ASR) toolkit (Povey et al., 2011) as its backend. Kaldi is a mature, widely used and well-supported speech recognition toolkit which supports a range of hidden Markov model based speech recognition models.

[1] https://github.com/CoEDL/elpis
[2] https://github.com/kaldi-asr/kaldi

In this paper we report on the ongoing integration of ESPnet[3] into Elpis as an alternative to the current Kaldi system. We opted to integrate ESPnet (Watanabe et al., 2018) as it is a widely used and actively developed tool with state-of-the-art end-to-end neural network models. By supporting ESPnet in Elpis, we aim to bring a wider range of advances in speech recognition to a broad group of users, and provide alternative model options that may better suit some data circumstances, such as an absence of a pronunciation lexicon.

In the rest of this paper, we describe changes to the Elpis toolkit to support the new backend, and preliminary experiments applying our ESPnet recipe to several datasets from a language documentation context. Finally, we discuss plans going forward with this project.

2 Related Work

Automatic phonetic/phonemic transcription in language documentation As a subset of speech recognition research, work has been done in applying speech recognition systems to the very low-resource phonemic data scenarios typical in the language documentation context. Encouraging results capitalizing on the advances in speech recognition technology for automatic phonemic transcription in a language documentation context were reported by Adams et al. (2018). Their work used a neural network architecture with connectionist temporal classification (Graves et al., 2006) for phonemic (including tonal) transcription. A command line toolkit was released called Persephone. To assess the reproducibility of the results on other languages, experiments were extended beyond the Chatino, Na and Tsuut'ina data sets, to a sample of languages from the Pangloss Collection, an online archive of under-resourced languages (Michailovsky et al., 2014). The results confirmed that end-to-end models for automatic phonemic transcription deliver promising performance, and also suggested that preprocessing tasks can to a large extent be automated, thereby increasing the attractiveness of the tool for language documentation workflows (Wisniewski et al., 2020). Another effort in this space is Allosaurus (Li et al., 2020), which leverages multilingual models for phonetic transcription and jointly models language independent phones and language-dependent phonemes. This stands as a promising step towards effective universal phonetic recognition, which would be of great value in the language documentation process.

User-friendly speech recognition interfaces Since such research tools do not have user friendly interfaces, efforts have been put into making these tools accessible to wider audience of users. The authors of Allosaurus provide a web interface online.[4] To integrate Persephone into the language documentation workflow, a plugin, Persephone-ELAN,[5] was developed for ELAN,[6] a piece of software that is widely used for annotation in language documentation (Cox, 2019).

Meanwhile, Elpis is a toolkit that provides a user-friendly front-end to the Kaldi speech recognition system. The interface steps the user through the process of preparing language recordings using existing ELAN transcription files, training a model and applying the model to obtain a hypothesis orthographic transcription for untranscribed speech recordings.

3 Bringing ESPnet to Elpis

ESPnet is an end-to-end neural network-based speech recognition toolkit. Developed with Pytorch (Paszke et al., 2019) in a research context, the tool satisfies three desiderata for our purposes: (a) it is easy to modify training *recipes*, which consist of collections of scripts and configuration files that make it easy to perform training and decoding by calling a wrapper script. These recipes describe a wide range of the hyperparameters and architecture choices of the model; (b) it is actively developed, with frequent integration of the latest advances in end-to-end speech recognition; and (c) it supports Kaldi-style data formatting, which makes it a natural end-to-end counterpart to Kaldi backend that was already supported in Elpis. These points make it a more appealing candidate backend than Persephone, primarily due to ESPnet's larger developer base.

3.1 Development of an ESPnet recipe for Elpis

One goal of the integration is to create a default ESPnet recipe for Elpis to use, that performs well across a variety of languages and with the small

[3] https://github.com/espnet/espnet

[4] https://www.dictate.app

[5] https://github.com/coxchristopher/persephone-elan

[6] https://archive.mpi.nl/tla/elan

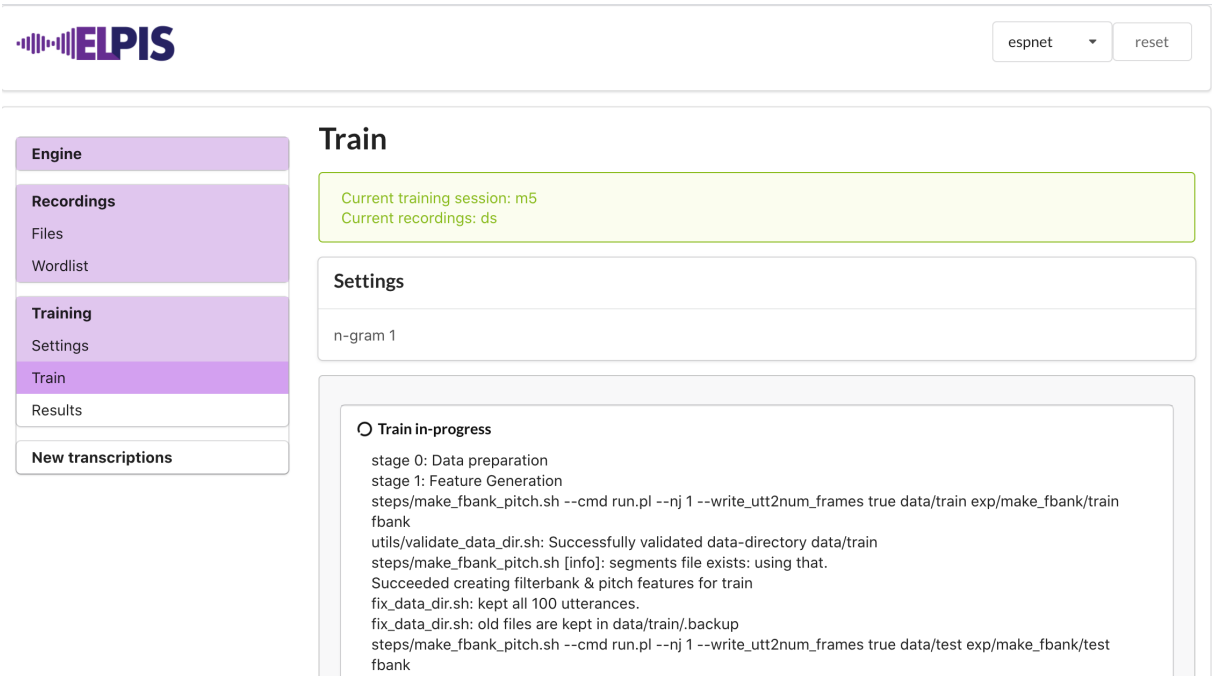

Figure 1: Training stages of the Elpis interface. Notice the choice of backend in the upper right-hand corner.

amount and type of data typically available in a language documentation context.

To get a sense of how easy it would be using ESPnet to get similar performance as previously attained we applied it to the single-speaker Na and Chatino datasets as used in Adams et al. (2018) (see Table 1, which includes other details of the datasets used, including the amount of training data). We report character error rate (CER) rather than phoneme error rate (PER) because it is general, does not require a subsequent language-specific post-processing step, and also captures characters that a linguist might want transcribed that aren't strictly phonemic. Because of minor differences in the training sets, their preprocessing, and metrics used, these numbers are not intended to be directly comparable with previous work. While these results are not directly comparable to the results they reported, the performance was good enough to confirm that integrating ESPnet was preferable to Persephone. We do no language-specific preprocessing, though the Elpis interface allows the user to define a character set for which instances of those characters will be removed from the text. For the Na data and the Japhug data in §4, the Pangloss XML format is converted to ELAN XML using a XSLT-based tool, Pangloss-Elpis[7].

While we did not aggressively tune hyperparameters and architecture details, they do have a substantial impact on performance and computational requirements. Owing to the small datasets and limited computational resources of many of the machines that Elpis may run on, we used a relatively small neural network. In the future we aim to grow a representative suite of evaluation languages from a language documentation setting for further tuning to determine what hyperparameters and architecture best suit different scenarios. Though we aim for a recipe that does well across a range of possible language documentation data circumstances, the best architecture and hyperparameters will vary depending on the characteristics of the input dataset. Rather than have the user fiddle with such parameters directly, which would undermine the user-friendliness of the tool, there is potential to automatically adjust the hyperparameters of the model on the basis of the data supplied to the model. For example, the parameters could be automatically set depending on the number of speakers in the ELAN file and the total amount of speech.

The architecture we used for these experiments is a hybrid CTC-attention model (Watanabe et al., 2017b) with a 3-layer BiLSTM encoder and a single layer decoder. We use a hidden size of 320

[7] https://gitlab.com/lacito/pangloss-elpis

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

Table 1: Information on the evaluation datasets used and the character error rate performance of the current recipe.

and use an equal weighting between the CTC and attention objectives. For optimization we use a batch length of 30 and the Adadelta gradient descent algorithm (Zeiler, 2012). For more details, we include a link to the recipe.[8]

3.2 Elpis enhancements

Beyond integration of ESPnet into Elpis, several other noteworthy enhancements have been made to Elpis.

Detailed training feedback Prior to the work reported in this paper, the progress of training and transcribing stages was shown as a spinning icon with no other feedback. Due to the amount of time it takes to train even small speech recognition models, the lack of detailed feedback may cause a user to wonder what stage the training was at, or whether a fault had caused the system to fail. During training and transcription, the backend processes' logs are now output to the screen (see Figure 1). Although the information in these logs may be more complex than what the intended audience of the tool understands, it does serve to give any user feedback on how training is going, and reassure them that it *is* still running (or notify them if a process has failed). The logs can also provide useful contextual information when debugging an experiment in collaborations between language workers and software engineers.

CUDA-supported Docker image The type of Kaldi model which Elpis trains was originally selected to be computationally efficient, and able to run on the type of computers commonly used by language researchers. With the addition of ESPnet, the benefit of using more computing power will be felt through reduced training times for the neural network. To this end, Elpis has been adapted to include Compute Unified Device Architecture (CUDA) support, which is essential in order to leverage a GPU when training ESPnet on a machine that has one available.

[8] https://github.com/persephone-tools/espnet/commit/1c529eab738cc8e68617aebbae520f7c9c919081

4 Application to a new data set: Japhug

The point of this work is to provide a tool that can be used by linguists in their limited-data scenarios. To this end we aim to experiment with diverse datasets that reflect the breadth of language documentation contexts. Going forward, this will be useful in getting a sense of what sort of model performance users can expect given the characteristics of dataset. In this section we report on further application of the model underpinning the Elpis–ESPnet integration to another data set.

Japhug is a Sino-Tibetan language with a rich system of consonant clusters, as well as flamboyant morphology. In Japhug, syllables can have initial clusters containing at most three consonants, and at most one coda (Jacques, 2019). Japhug does not have lexical tones. The language's phonological profile is thus very different from Na (about which see Michaud, 2017) and Chatino (Cruz, 2011; Cruz and Woodbury, 2014; Cavar et al., 2016).

The data set comprises a total of about 30 hours of transcribed recordings of narratives, time-aligned at the level of the sentence, which is a huge amount in a language documentation context. The recordings were made in the course of field trips from the first years of the century until now, in a quiet environment, and almost all of a single speaker. Our tests on various data sets so far suggest that these settings (one speaker – hence no speaker overlap – and clean audio) are those in which performance is most likely to be good when one happens to be training an acoustic model from scratch.

The full data set is openly accessible online from the Pangloss Collection, under a Creative Commons license, allowing visitors to browse the texts, and computer scientists to try their hand at

the data set.[9] The data collector's generous approach to data sharing sets an impressive example, putting into practice some principles which gather increasing support, but which are not yet systematically translated into institutional and editorial policies (Garellek et al., 2020).

The dataset can be downloaded by sending a request to the Cocoon data repository, which hosts the Pangloss Collection. A script, *retriever.py*,[10] retrieves resources with a certain language name. Data sets can then be created in various ways, such as sorting by speaker (tests suggest that single-speaker models are a good way to start) and by genre, e.g. excluding materials such as songs, which are a very different kettle of fish from ordinary speech and complicate model training.

Figure 2 shows how the phoneme error rate decreases as the amount of training data increases up to 170 minutes. Tests are currently being conducted to verify whether performance stagnates when the amount of data is increased beyond 170 minutes. As with the other experiments the recipe described in §3.1 was used. For each amount of training data, the model was trained for 20 epochs for each of these training runs, with the smaller sets always as a subset of all larger sets. Figure 3 shows the training profile for a given training run using 170 minutes of data.

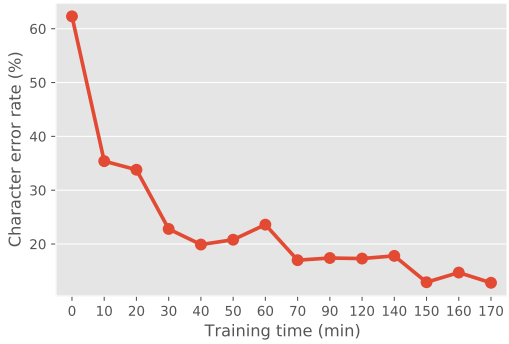

Figure 2: Character error rate for Japhug as a function of the amount of training data, using the ESPnet recipe included in Elpis.

[9]Each text has a Digital Object Identifier, allowing for one-click access. Readers are invited to take a look: https://doi.org/10.24397/pangloss-0003360

[10]https://gitlab.com/lacito/pangloss-elpis

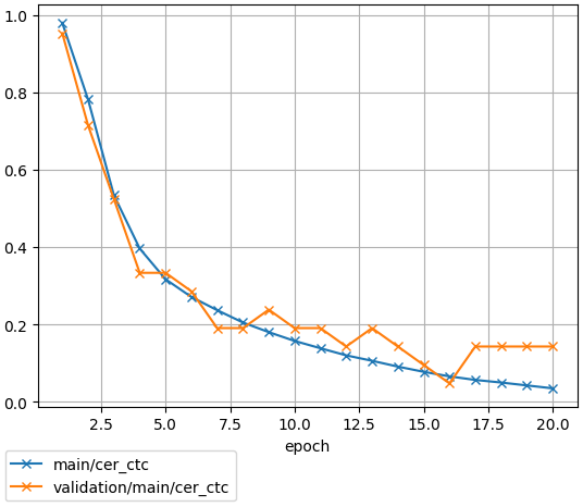

Figure 3: Character error rate on the training set (blue) and validation set (orange) for Japhug as training progresses (up to 20 epochs), using the ESPnet recipe included in Elpis.

5 Challenges concerning adoption of automatic speech recognition tools in language documentation

Devoting a section to reflections about adoption of automatic speech recognition tools in language documentation may seem superfluous here. The audience of a conference on the use of computational methods in the study of endangered languages is highly knowledgeable about the difficulties and the rewards of interdisciplinary projects, as a matter of course. But it seemed useful to include a few general thoughts on this topic nonetheless, for the attention of the broader readership which we hope will probe into the Proceedings of the ComputEL-4 conference: colleagues who may consider joining international efforts for wider adoption of natural language processing tools in language documentation workflows. We briefly address a few types of doubts and misgivings.

5.1 Is automatic speech recognition software too complex for language workers?

A first concern is that automatic speech recognition software is simply too complex for language workers. But it should be recalled that new technologies that seem inaccessible to language workers can be game-changers in linguistics. For instance, the LaTeX software is the typesetting backend used by the journal *Glossa* (Rooryck, 2016) and by the publishing house Language Science Press (Nordhoff, 2018), which publish research in linguistics, offering high-quality open-access

venues with no author fees or reader fees. Thus, LaTeX, a piece of software which is notorious for its complexity, is used on a large scale in linguistics publishing: *Glossa* publishes more than 100 articles a year, and Language Science Press about 30 books a year. Key to this success is an organizational setup whereby linguists receive not only a set of stylesheets and instructions, but also hands-on support from a LaTeX expert all along the typesetting process. Undeniably complex software is only accessible to people with no prior knowledge of it if support is available. Automatic speech recognition software should be equally accessible for language workers, given the right organization and setup. Accordingly, special emphasis is placed on user design in the Elpis project. This aspect of the work falls outside of the scope of the present paper, but we wanted to reassure potential users that it is clear to Elpis developers that the goal is to make the technology available to people who do not use the command line. If users can operate software such as ELAN then they will be more than equipped for the skills of uploading ELAN files to Elpis and clicking the Train button.

5.2 Will the technology deliver on its promise?

A second concern among language workers is whether the technology can deliver on its promise, or whether transcription acceleration projects are a case of "digital innovation fetishism" (Ampuja, 2020). Some language workers have reported a feeling that integration of automatic transcription into the language documentation workflow (as described in Michaud et al., 2018) feels out of reach for them. There is no denying that natural language processing tools such as ESPnet and Kaldi are very complex, and that currently, the help of specialists is still needed to make use of this technology in language documentation. However, progress is clearly being made, and a motivated interdisciplinary community is growing at the intersection of language documentation and computer science, comprising linguists who are interested in investing time to learn about natural language processing and computer scientists who want to achieve "great things with small languages", in Nick Thieberger's phrase (Thieberger and Nordlinger, 2006). It seems well worth investing in computational methods to assist in the urgent task of documenting the world's languages.

5.3 Keeping up with the state of the art vs. stabilizing the tool

Finally, a concern among linguists is that the state of the art in computer science is evolving so rapidly that the tool cannot be stabilized, and hence cannot be proposed to language workers for enduring integration into the language documentation workflow. In cases where significant, high-frequency updates are required to keep up with changes in speech recognition software, the investment could be too much for the relatively small communities of programmers involved in transcription acceleration projects.

Our optimistic answer is that state-of-the-art code, or code close to the state of the art, need not be difficult to integrate, use or maintain. For example, the developers of Huggingface's Transformers [11] do an impressive job of wrapping the latest and greatest in natural language processing into an easy-to-use interface (Wolf et al., 2019). They have shown an ability to integrate new models quickly after their initial publication. Usability and stability of the interface is dictated by the quality of the code that is written by the authors of the backend library. If this is done well then the state of the art can be integrated with minimal coding effort by users of the library. For this reason, we are not so concerned about the shifting sands of the underlying building blocks, but the choice of quality backend library does count here. It is also true that there will have to be some modest effort to keep up to date with ESPnet – as would be the case using any other tool.

6 Further improvements

The broader context to the work reported here is a rapidly evolving field in which various initiatives aim to package natural language processing toolkits in intuitive interfaces so as to allow a wider audience to leverage the power of these toolkits. Directions for new developments in Elpis include (i) refining the ESPnet recipe, (ii) refining the user interface through user design processes, (iii) preparing pre-trained models that can be adapted to a small amount of data in a target language, and (iv) providing Elpis as a web service.

[11] https://github.com/huggingface/transformers

6.1 Refining the ESPnet recipe

Refinement of the ESPnet recipe that is used in the Elpis pipeline, such that it works as well as possible given the type of data found in language documentation contexts, is a top priority. This work focuses on achieving lower error rates across data sets, starting with refining hyperparameters for model training and extends to other project objectives including providing pre-trained models (see §6.3). This work is of a more experimental nature and can be done largely independently of the Elpis front-end.

6.2 Refining the interface

In parallel with the technical integration of ESPnet with Elpis, a user-design process has been investigating how users expect to use these new features. In a series of sessions, linguists and language workers discussed their diverse needs with a designer. The feedback from this process informed the building of a prototype interface based on the latest version of Elpis at the time. The test interface was then used in individual testing sessions to discover points of confusion and uncertainty in the interface. Results of the design process will guide an update to the interface and further work on writing supporting documentation and user guides. The details of this process are beyond the scope of this paper and will be reported separately in future.

6.3 Pre-trained models and transfer learning

Adapting a trained model to a new language has a long history in speech recognition, having been used both for Hidden Markov Model based systems (Schultz and Waibel, 2001; Le and Besacier, 2005; Stolcke et al., 2006; Tóth et al., 2008; Plahl et al., 2011; Thomas et al., 2012; Imseng et al., 2014; Do et al., 2014; Heigold et al., 2013; Scharenborg et al., 2017) and end-to-end neural systems (Toshniwal et al., 2017; Chiu et al., 2018; Müller et al., 2017; Dalmia et al., 2018; Watanabe et al., 2017a; Inaguma et al., 2018; Yi et al., 2018; Adams et al., 2019). In scenarios where data in the target domain or language is limited, leveraging models trained on a number of speakers in different languages often can result in a better performance. The model can learn to cope with acoustic and phonetic characteristics that are common between languages, such as building robustness to channel variability due to different recording conditions, as well as learning common features of phones and sequences of phones between languages.

In recent years pre-training of models on large amounts of *unannotated* data has led to breakthrough results in text-based natural language processing, initially gaining widespread popularity with the context-independent embeddings of word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), before the recent contextual word embedding revolution (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2020) that has harnessed the transformer architecture (Vaswani et al., 2017). It is now the case that the best approaches in natural language processing are typically characterized by pre-training of a model on a large amount of unannotated data before fine-tuning to a target task, which typically involves the cloze task (masked language modeling). Models pre-trained in this way best make use of available data since unannotated data far outweighs annotated data and such pre-training is advantageous to downstream learning, whether a small or large amount of data is available in the target task (Gururangan et al., 2020). Despite the established nature of pre-training in natural language processing, it is less well established in speech recognition, though there has been recent work (Rivière et al., 2020; Baevski et al., 2020).

The language documentation scenario, where annotated data is very limited is a scenario that we argue stands most to gain from such pre-training (both supervised and self-supervised out-of-domain); followed by model adaptation to limited target language data. One of the features Elpis could provide is to include pre-trained models in its distribution or via an online service. Such models may be pre-trained in a self-supervised manner on lots of untranscribed speech, trained in a supervised manner on transcribed speech in other languages, or use a combination of both pre-training tasks. In cases where the pre-trained model was trained in a supervised manner, there is scope to deploy techniques to reconcile the differences in acoustic realization between phonemes of different languages via methods such as that of Allosaurus (Li et al., 2020) which uses a joint model of language-independent phones and language-dependent phonemes. Providing a variety of pre-trained models would be valuable, since the best seed model for adaptation may vary on the basis

of the data in the target language (Adams et al., 2019).

A recognized problem in language documentation is that, owing to the transcription bottleneck, a large amount of unannotated and untranscribed data ends up in *data graveyards* (Himmelmann, 2006): archived recordings that go unused in linguistic research. It is frequently the case that the vast majority of speech collected by field linguists is untranscribed. Here too, self-supervised pre-training in the target language is likely a promising avenue to pursue, perhaps in tandem with supervised pre-training regiments. For this reason, we are optimistic that automatic transcription will have a role to play in almost all data scenarios found in the language documentation context – even when training data is extremely limited – and are not just reserved for certain single-speaker corpora with consistently high quality audio and clean alignments with text. In the past one could plausibly argue that the limited amount of transcribed speech as training data is an insurmountable hurdle in a language documentation context, but that will likely not remain the case.

One of the next steps planned for Elpis is to allow for acoustic models to be exported and loaded. Beyond the immediate benefit of saving the trouble of training models anew each time, having a library of acoustic models available in an online repository would facilitate further research on adaptation of acoustic models to (i) more speakers, and (ii) more language varieties. Building universal phone recognition systems is an active area of research (Li et al., 2020); these developments could benefit from the availability of acoustic models on a range of languages. Hosting acoustic models in an online repository, and using them for transfer learning, appear as promising perspectives.

6.4 Providing Elpis as a web service

Training models requires a lot of computing power. Elpis now supports high-speed parallel processing in situations where the user's operating system has compatible GPUs (graphics processing units) (see Section 3.2). However, many users don't have this technology in the computers they have ready access to, so we also plan to investigate possibilities for hosting Elpis on a high-capacity server for end-user access. Providing language technologies via web services appears to be a successful method of making tools widely available, with examples including the WebMAUS forced-alignment tool.[12] The suite of tools provided by the Bavarian Speech Archive (Kisler et al., 2017) have successfully processed more than ten million media files since their introduction in 2012. For users who want to avoid sending data to a server, there are other possibilities: Kaldi can be compiled to Web Assembly so it can do decoding in a browser (Hu et al., 2020). But for the type of user scenarios considered here, hosting on a server would have major advantages, and transfer over secure connection is a strong protection against data theft (for those data sets that must not be made public, to follow the consultants' wishes or protect the data collectors' exclusive access rights to the data so that they will not be scooped in research and placed at a disadvantage in job applications).

This context suggests that it would be highly desirable to design web hosting for Elpis. It would facilitate conducting broad sets of tests training acoustic models, and would also facilitate the transcription of untranscribed recordings.

7 Conclusion

In this paper we have reported on integrating ESPnet, an end-to-end neural network speech recognition system, into Elpis, the user-friendly speech recognition interface. We described changes that have been made to the front-end, the addition of a CUDA supported Elpis Dockerfile, and the creation of an ESPnet recipe for Elpis. We reported preliminary results on several languages and articulated plans going forward.

Acknowledgments

Many thanks to the three reviewers for comments and suggestions.

We are grateful for financial support to the Elpis project from the Australian Research Council Centre of Excellence for the Dynamics of Language, the University of Queensland, the *Institut des langues rares* (ILARA) at *École Pratique des Hautes Études*, the European Research Council (as part of the project "Beyond boundaries: Religion, region, language and the state" [ERC-609823]) and *Agence Nationale de la Recherche* (as part of two projects, "Computational Language

[12] https://clarin.phonetik.uni-muenchen.de/BASWebServices/interface

Documentation by 2025" [ANR-19-CE38-0015-04] and "Empirical Foundations of Linguistics" [ANR-10-LABX-0083]).

Linguistic resources used in the present study were collected as part of projects funded by the European Research Council ("Discourse reporting in African storytelling" [ERC-758232]) and by *Agence Nationale de la Recherche* ("Parallel corpora in languages of the Greater Himalayan area" [ANR-12-CORP-0006]).

Many thanks to Cécile Macaire for adding time codes to the Japhug annotation (during an internship at LACITO and LIG in 2020), allowing for use of the data for training acoustic models.

References

Oliver Adams, Trevor Cohn, Graham Neubig, Hilaria Cruz, Steven Bird, and Alexis Michaud. 2018. Evaluating phonemic transcription of low-resource tonal languages for language documentation. In *Proceedings of LREC 2018 (Language Resources and Evaluation Conference)*, pages 3356–3365, Miyazaki. https://halshs.archives-ouvertes.fr/halshs-01709648.

Oliver Adams, Matthew Wiesner, Shinji Watanabe, and David Yarowsky. 2019. Massively multilingual adversarial speech recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 96–108, Minneapolis, Minnesota. Association for Computational Linguistics. https://www.aclweb.org/anthology/N19-1009.

Marko Ampuja. 2020. The blind spots of digital innovation fetishism. In Matteo Stocchetti, editor, *The digital age and its discontents: Critical reflections in education*, pages 31–54. Helsinki University Press, Helsinki. https://doi.org/10.33134/HUP-4-2.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*. https://arxiv.org/abs/2006.11477.

Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. 2014. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56:85–100.

Rogier Blokland, Marina Fedina, Ciprian Gerstenberger, Niko Partanen, Michael Rießler, and Joshua Wilbur. 2015. Language documentation meets language technology. In *Proceedings of the First International Workshop on Computational Linguistics for Uralic Languages - Septentrio Conference Series*, pages 8–18. http://septentrio.uit.no/index.php/SCS/article/view/3457/3386.

Małgorzata Cavar, Damir Cavar, and Hilaria Cruz. 2016. Endangered language documentation: bootstrapping a Chatino speech corpus, forced aligner, ASR. In *Proceedings of LREC 2016, Tenth International Conference on Language Resources and Evaluation*, pages 4004–4011, Portorož, Slovenia.

Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Katya Gonina, et al. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *ICASSP*, pages 4774–4778. https://arxiv.org/abs/1712.01769.

Christopher Cox. 2019. Persephone-ELAN (software). https://github.com/coxchristopher/persephone-elan. https://github.com/coxchristopher/persephone-elan.

Emiliana Cruz. 2011. *Phonology, tone and the functions of tone in San Juan Quiahije Chatino*. Ph.{d}., University of Texas at Austin, Austin.

Emiliana Cruz and Tony Woodbury. 2014. Finding a way into a family of tone languages: The story and methods of the Chatino Language Documentation Project. *Language Documentation and Conservation*, 8:490–524.

Siddharth Dalmia, Ramon Sanabria, Florian Metze, and Alan W Black. 2018. Sequence-based multilingual low resource speech recognition. In *ICASSP*. https://arxiv.org/abs/1802.07420.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. https://www.aclweb.org/anthology/N19-1423.

Van Hai Do, Xiong Xiao, Eng Siong Chng, and Haizhou Li. 2014. Cross-lingual phone mapping for large vocabulary speech recognition of under-resourced languages. *IEICE Transactions on Information and Systems*, E97-D(2):285–295.

Daan van Esch, Ben Foley, and Nay San. 2019. Future directions in technological support for language documentation. In *Proceedings of the Workshop on Computational Methods for Endangered Languages*, volume 1, Honolulu, Hawai'i. https://computel-workshop.org/wp-content/uploads/2019/02/CEL3_book_papers_draft.pdf#page=26.

Ben Foley, Josh Arnold, Rolando Coto-Solano, Gautier Durantin, and T. Mark Ellison. 2018. Building speech recognition systems for language documentation: the CoEDL Endangered Language Pipeline and Inference System (ELPIS). In *Proceedings of the 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU), 29-31 August 2018*, pages 200–204, Gurugram, India. ISCA. https://www.isca-speech.org/archive/SLTU_2018/pdfs/Ben.pdf.

Ben Foley, Alina Rakhi, Nicholas Lambourne, Nicholas Buckeridge, and Janet Wiles. 2019. Elpis, an accessible speech-to-text tool. In *Proceedings of Interspeech 2019*, pages 306–310, Graz. https://www.isca-speech.org/archive/Interspeech_2019/pdfs/8006.pdf.

Marc Garellek, Matthew Gordon, James Kirby, Wai-Sum Lee, Alexis Michaud, Christine Mooshammer, Oliver Niebuhr, Daniel Recasens, Timo Roettger, Adrian Simpson, and Kristine M. Yu. 2020. Toward open data policies in phonetics: What we can gain and how we can avoid pitfalls. *Journal of Speech Science*, 9(1). https://halshs.archives-ouvertes.fr/halshs-02894375.

Alex Graves, Santiago Fernandez, Faustino Gomez, and Jurgen Schmidhuber. 2006. Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks. *Proceedings of the 23rd international conference on Machine Learning*, pages 369–376. http://www.cs.utoronto.ca/~graves/icml_2006.pdf.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360. Association for Computational Linguistics. https://arxiv.org/abs/2004.10964.

Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur. 2018. End-to-end speech recognition using lattice-free MMI. In *Interspeech*, pages 12–16. https://danielpovey.com/files/2018_interspeech_end2end.pdf.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*. https://arxiv.org/abs/1412.5567.

Georg Heigold, Vincent Vanhoucke, Alan Senior, Patrick Nguyen, Marc'Aurelio Ranzato, Matthieu Devin, and Jeffrey Dean. 2013. Multilingual acoustic models using distributed deep neural networks. In *Proceedings of ICASSP*, pages 8619–8623.

Nikolaus Himmelmann. 2006. Language documentation: what is it and what is it good for? In Josh Gippert, Nikolaus Himmelmann, and Ulrike Mosel, editors, *Essentials of language documentation*, pages 1–30. de Gruyter, Berlin/New York.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and Others. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97.

Nils Hjortnaes, Niko Partanen, Michael Rießler, and Francis M. Tyers. 2020. Towards a speech recognizer for Komi, an endangered and low-resource Uralic language. In *Proceedings of the Sixth International Workshop on Computational Linguistics of Uralic Languages*, pages 31–37, Wien. Association for Computational Linguistics. https://www.aclweb.org/anthology/2020.iwclul-1.5/.

Mathieu Hu, Laurent Pierron, Emmanuel Vincent, and Denis Jouvet. 2020. Kaldi-web: An installation-free, on-device speech recognition system. In *Proceedings of Interspeech 2020 Show & Tell*, Shanghai. https://hal.archives-ouvertes.fr/hal-02910876.

David Imseng, Petr Motlicek, Hervé Bourlard, and Philip N Garner. 2014. Using out-of-language data to improve an under-resourced speech recognizer. *Speech Communication*, 56:142–151.

Hirofumi Inaguma, Jaejin Cho, Murali Karthick Baskar, Tatsuya Kawahara, and Shinji Watanabe. 2018. Transfer learning of language-independent end-to-end ASR with language model fusion. *arXiv:1811.02134*. https://arxiv.org/abs/1811.02134.

Guillaume Jacques. 2019. Japhug. *Journal of the International Phonetic Association*, 49(3):427–450.

Thomas Kisler, Uwe Reichel, and Florian Schiel. 2017. Multilingual processing of speech via web services. *Computer Speech & Language*, 45:326–347. ISBN: 0885-2308 Publisher: Elsevier.

Viet Bac Le and Laurent Besacier. 2005. First steps in fast acoustic modeling for a new target language: application to Vietnamese. In *ICASSP*.

Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anastasopoulos, David R. Mortensen, Graham Neubig, and Alan W. Black. 2020. Universal phone recognition with a multilingual allophone system. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8249–8253. IEEE. https://arxiv.org/abs/2002.11800.

KyungTae Lim, Niko Partanen, and Thierry Poibeau. 2018. Multilingual dependency parsing for low-resource languages: Case studies on North Saami and Komi-Zyrian. In *Proceedings of LREC (International Conference on Language Resources and Evaluation)*, Miyazaki. https://hal.archives-ouvertes.fr/hal-01856178.

Qi Liu, Matt J Kusner, and Phil Blunsom. 2020. A survey on contextual embeddings. *arXiv preprint arXiv:2003.07278*. https://arxiv.org/abs/2003.07278.

Boyd Michailovsky, Martine Mazaudon, Alexis Michaud, Séverine Guillaume, Alexandre François, and Evangelia Adamou. 2014. Documenting and researching endangered languages: the Pangloss Collection. *Language Documentation and Conservation*, 8:119–135. https://halshs.archives-ouvertes.fr/halshs-01003734.

Alexis Michaud. 2017. *Tone in Yongning Na: lexical tones and morphotonology*. Number 13 in Studies in Diversity Linguistics. Language Science Press, Berlin. http://langsci-press.org/catalog/book/109.

Alexis Michaud, Oliver Adams, Trevor Cohn, Graham Neubig, and Séverine Guillaume. 2018. Integrating automatic transcription into the language documentation workflow: experiments with Na data and the Persephone toolkit. *Language Documentation and Conservation*, 12:393–429. http://hdl.handle.net/10125/24793.

Tomáš Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Markus Müller, Sebastian Stüker, and Alex Waibel. 2017. Phonemic and graphemic multilingual CTC based speech recognition. *arXiv:1711.04564*. https://arxiv.org/abs/1711.04564.

Sebastian Nordhoff. 2018. *Language Science Press business model: Evaluated version of the 2015 model*. Language Science Press, Berlin. https://doi.org/10.5281/zenodo.1286972.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, and Luca Antiga. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems. Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pages 8026–8037, Vancouver, Canada. https://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*. https://arxiv.org/abs/1802.05365.

Christian Plahl, Ralf Schlüter, and Hermann Ney. 2011. Cross-lingual portability of Chinese and English neural network features for French and German LVCSR. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 371–376.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. https://infoscience.epfl.ch/record/192584/files/Povey_ASRU2011_2011.pdf.

Mirco Ravanelli, Titouan Parcollet, and Yoshua Bengio. 2019. The pytorch-kaldi speech recognition toolkit. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6465–6469. IEEE. https://arxiv.org/abs/1811.07453.

Morgane Rivière, Armand Joulin, Pierre-Emmanuel Mazaré, and Emmanuel Dupoux. 2020. Unsupervised pretraining transfers well across languages. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7414–7418. https://arxiv.org/abs/2002.02848.

Johan Rooryck. 2016. Introducing *Glossa*. *Glossa*, 1(1):1–3. http://doi.org/10.5334/gjgl.91.

Odette Scharenborg, Francesco Ciannella, Shruti Palaskar, Alan Black, Florian Metze, Lucas Ondel, and Mark Hasegawa-Johnson. 2017. Building an ASR system for a low-research language through the adaptation of a high-resource language ASR system: preliminary results. In *International Conference on Natural Language, Signal and Speech Processing (ICNLSSP)*.

Tanja Schultz and Alex Waibel. 2001. Experiments on cross-language acoustic modeling. *EUROSPEECH'01*, pages 2721–2724.

Andreas Stolcke, Frantisek Grezl, Mei-Yuh Hwang, Xin Lei, Nelson Morgan, and Dimitra Vergyri.

2006. Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons. In *ICASSP*. http://ieeexplore.ieee.org/document/1660022/.

Nick Thieberger and Rachel Nordlinger. 2006. Doing great things with small languages (Australian Research Council grant DP0984419). https://arts.unimelb.edu.au/school-of-languages-and-linguistics/our-research/past-research-projects/great-things-small-languages.

Samuel Thomas, Sriram Ganapathy, Hynek Hermansky, and Speech Processing. 2012. Multilingual MLP features for low-resource LVCSR systems. In *ICASSP*, pages 4269–4272.

Shubham Toshniwal, Tara N. Sainath, Ron J. Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao. 2017. Multilingual speech recognition with a single end-to-end model. In *ICASSP*. http://arxiv.org/abs/1711.01694.

László Tóth, Joe Frankel, Gábor Gosztolya, and Simon King. 2008. Cross-lingual portability of MLP-based tandem features - a case study for English and Hungarian. *INTERSPEECH*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Shinji Watanabe, Takaaki Hori, and John R Hershey. 2017a. Language independent end-to-end architecture for joint language identification and speech recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 265–271.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, and Nanxin Chen. 2018. ESPnet: End-to-end speech processing toolkit. *arXiv preprint arXiv:1804.00015*. https://arxiv.org/abs/1804.00015.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. 2017b. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253.

Guillaume Wisniewski, Séverine Guillaume, and Alexis Michaud. 2020. Phonemic transcription of low-resource languages: To what extent can preprocessing be automated? In *Proceedings of the 1st Joint SLTU (Spoken Language Technologies for Under-resourced languages) and CCURL (Collaboration and Computing for Under-Resourced Languages) Workshop*, pages 306–315, Marseille, France. European Language Resources Association (ELRA). https://halshs.archives-ouvertes.fr/hal-02513914.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Jiangyan Yi, Jianhua Tao, Zhengqi Wen, and Ye Bai. 2018. Adversarial multilingual training for low-resource speech recognition. *ICASSP*, pages 4899–4903.

Matthew D. Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*. https://arxiv.org/abs/1212.5701.

Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney. 2018. Improved training of end-to-end attention models for speech recognition. https://arxiv.org/abs/1805.03294.

Wei Zhou, Wilfried Michel, Kazuki Irie, Markus Kitza, Ralf Schlüter, and Hermann Ney. 2020. The RWTH ASR system for TED-LIUM Release 2: Improving Hybrid HMM with SpecAugment. https://arxiv.org/abs/2004.00960.