Generalized Lexicon and Translation Model Construction for Endangered Language Transcription

Riya Patel **Swetha Gangu** **Nathaniel Robinson** **Graham Neubig**

Carnegie Mellon University
{riyap,sgangu,nrrobins,gneubig}@andrew.cmu.edu

Abstract

For endangered languages, data is scarce, and lexicons are often not available, which makes it difficult to develop automatic transcription systems. WFST-based methods leverage phoneme lattices and translations of low-resource language speech into higher-resourced languages to improve transcription. Though intended for endangered languages, these methods were only applied to Japanese in prior work. Using recent developments in language-universal automatic speech recognition (ASR), we extend these methods to a language-universal setting. We present a language-agnostic pipeline for learning transcriptions and show results for a variety of languages. Our findings suggest that the WFST-based approach is in fact counterproductive in some settings. We discuss ways this could be mitigated.

1 Introduction and Problem Statement

Language documentation is necessary for preserving endangered languages with little to no resources. Speech data for these languages is often relatively easy to collect, but transcribing it is prohibitively slow. Michaud *et al.* (2014) estimated that it takes a trained linguist roughly 1 hour to transcribe 1 minute of speech. Language documentation and revitalization efforts can be aided greatly by automatic transcription systems.

We build on Adams *et al.*'s (2016b) approach to improve ASR by learning a latent lexicon and translation model from bilingual data. Adams *et al.*'s formulation requires phoneme lattices for input audio. Because these lattices can be hard to obtain for endangered and low-resource languages, they only applied their approach to high-resource Japanese. Using a language-universal phone recognizer [Li *et al.*, 2020], we generalize this approach for rapid adaption to endangered languages. We contribute:

- A language-agnostic ASR pipeline that does not require phoneme lattices
- An application of this pipeline to endangered Tulu
- Analysis of this method's failures in language universal settings

1.1 Motivation: The Case of Tulu

The endangered language Tulu could benefit from language universal ASR technologies. One of the UNESCO "Atlas of the World's Languages in Danger,"[1] Tulu is spoken in the states of Karnataka and Kerala in Southern India and has approximately 1.7 million speakers according to a 2001 census. UNESCO considers the degree of endangerment for Tulu to be "vulnerable", meaning that most children speak the language but usage is typically restricted to the home. Tulu is not taught in academic settings and its written script, Tigalari, is no longer in use. As a result, there is a high risk of Tulu becoming extinct in the coming decades. Improved lexical documentation of Tulu and translation models to the lingua franca can be useful as a learning resource to strengthen knowledge of Tulu in regions where it was originally spoken.

Language extinction represents a massive and irretrievable loss of culture, knowledge, and history. This work aims to develop transcription for languages at risk for extinction like Tulu. Tulu is an interesting case because it has a large number of native speakers, and there are multiple Tulu movies with English subtitles available on the web.

2 Related Work

We surveyed a number or related works pertaining to phoneme recognition via WFST models. We also review related work on Tulu language processing and methods.

[1] https://en.unesco.org/news/new-updates-available-interactive-atlas-world-s-languages-danger

2.1 ASR Via Translation and Lexicon Learning

Adams *et al.* (2016b) developed an ASR pipeline, LatticeTM, that learns transcription phonemes given audio data and transcribed translations. Their Bayesian non-parametric model expressed as a weighted finite-state transducer (WFST) predicts phonemes and latent textual transcriptions. LatticeTM approach models three distributions and maximizes their product:

$$\hat{\phi} = \arg\max_{\phi,f} P(x|\phi)P(\phi|f)P(f|e) \quad (1)$$

where ϕ is transcription phonemes, x is reference audio, f is latent textual transcription, and e is translation text. $\hat{\phi}$ is the set of phonemes predicted to match x. This method of phoneme prediction using translation data improved word error rate (WER) over ablations. The authors applied the framework for Japanese ASR but did not apply it to any endangered languages. And because they do not apply the approach with a universal phone recognizer, their method is not truly generalizable. To address this limitation, we leverage recent work in language-universal phone-based ASR to apply this strategy to endangered languages.

Other researchers have explored phoneme translations modeling [Duong *et al.*, 2016], computer-aided speech translation [Pelemans *et al.*, 2015], translation modeling from automatically transcribed speech [Paulik and Waibel, 2013], Bayesian word alignment [Mermer and Saraclar, 2011; Zezhong *et al.*, 2013], and language model learning from lattices [Neubig *et al.*, 2012].

There has also been extensive work on utilizing translation models to improve ASR performance [Vidal *et al.*, 2006]. However, this prior work does not use phoneme or phone lattices. Other previous work with lattices includes research to learn translation models from ASR word path lattices via WFSTs [Adams *et al.*, 2016a]. Our work will extend this vein of research by incorporating newer methods for phone recognition.

2.2 Preliminary Experiment

We used the ASR pipeline constructed by Adams *et al.* (2016b) (see equation (1)) and applied the phoneme recognition method to word lattices. We used the C++ implementation available on GitHub[2] and the OpenFST library[3]. Since the original dataset used in the paper was not made publicly available, we used the Fisher data set containing Spanish word lattices and their corresponding Spanish sentences and English translations, which is the same data set used for word prediction by Adams *et al.* (2016a). The Spanish lattices in the corpus were given in Joshua lattice format and thus had to be converted to OpenFST lattice format to be made compatible with the pipeline. We manually aligned the validation set and part of the test set for use in experimentation.

Aligned pairs	*WER*
100	53.4%
300	57.0%
700	47.9%
1200	49.1%
1700	46.3%
2500	61.9%

Table 1: Word error rate for ASR using different amounts of training data. Aligned pairs are Spanish audio mapped to English transcriptions, with word lattices given. Output sentences were normalized via Unicode.

Results for our preliminary work are in Table 1. Passing more data into the pipeline generally decreased WER. This model seems to perform fairly well in low-resource conditions. Smaller sentences seemed to be more accurately constructed compared to longer, more complex sentences. The exception to this is with spelling errors where the predicted "yea" would give a high error rate when compared with the Spanish translation "ya", and "muy" would give an error when compared with the original elaborated "muyyy". Conjugational differences also led to increased word error rates. Potentially, a different evaluation metric would be more apt to analyze the quality of the constructed sentence or running for more epochs or using larger datasets.

2.3 Universal Phone Recognition

Recent universal phone recognition methods allow us to leverage pre-trained models for low-resource applications, exploiting transfer learning from higher-resourced languages. The universal phone recognition model with multilingual allophone system by Li *et al.* (2020), or Allosaurus, improved performance by 17% phoneme error rate over the baseline conditions for unseen low-resourced testing languages. We use this technology to expand the LatticeTM approach to be

[2]https://github.com/oadams/latticetm
[3]https://www.openfst.org/

language-agnostic and not dependent on phoneme lattices.

2.4 Tulu Language Models

There has been limited research done on Tulu language, especially since it is endangered and has very limited resources. Antony *et al.* (2012) completed previous work in constructing a morphological analyzer and generator (MAG) for Tulu using AT&T OpenFST[4]. Their work focused on using a rules-based approach since there does not exist a large aligned, well-generated corpus to perform corpus-based methods of MAG. The MAG proposed in this work may be used to help in machine translation between Tulu to a more documented language like Kannada.

Shivakumar (2010) worked on expanding IndoWordNet to create a WordNet for the Tulu language. Since there were no previous lexical studies in Tulu, they used Hindi WordNet as a baseline. Resources were limited to a few dictionaries mapping Tulu to English, Tulu to Kannada, and English to Kannada, and two corpora of Tulu folk songs and tales with English translation. As a result, when creating the WordNet, Shivakumar gave equivalents in both Tulu and Kannada to allow for cross-checking when recognizing complex lexical units and compounds.

Both of these works rely on transcribed Tulu and suffered from limited availability of transcriptions. Our work on Tulu ASR can open up immensely greater possibilities for transcribed Tulu training data in order to improve applications such as MAG and lexicon documentation.

3 Methodology

We extend the approach of ASR improvement via latent lexicon and translation model learning with WFST to a language-agnostic setting.

3.1 Data Collection

Retrieving Tulu data was difficult since there are no public datasets of Tulu audio and corresponding translations to a higher resource language (English or Kannada). As a result, we created a dataset using Tulu movies (i.e. Kudla Cafe[5]) with corresponding English subtitles found on YouTube. Since all of the movies we found had burned-in subtitles, we extracted the subtitles and the timestamps during which they occurred on the screen using the Tesseract OCR Engine with a Python script[6]. We then cut the movie into audio clips based on the corresponding timestamps and performed noise gating techniques to remove background music and noise from each of the audio clips with a script using ffmpeg and sox[7].

In addition to testing our model's performance on Tulu, we tested the performance on several other languages as well, including Arabic, Spanish, Indonesian, Persian, and Tamil (a Dravidian language like Tulu). These languages are higher-resource than Tulu, so we were able to use HuggingFace's CoVoST 2 dataset[8], which contains audio recordings, transcriptions, and English translations for those languages. The audio recordings come from Mozilla's open-source Common Voice database[9] of crowdsourced voice recordings.

3.2 Transcription Pipeline

Our approach uses the same formulation described in equation (1), but our model does not require phoneme lattices. Instead we model $P(x|\phi)$ using language-universal phone recognizer, Allosaurus [Li *et al.*, 2020]. (See section 2.3.) Note that in this formulation ϕ represents language-agnostic phones rather than language-specific phonemes.

We represent Allosaurus outputs as phone lattices by setting $top\_k > 1$ and mapping multiple phone paths from state to state. See Figure 1. Because Allosaurus phone outputs are conditionally independent, there are no skip connections in lattices.

Our full pipeline downloads audio, translations, and transcriptions from HuggingFace and Mozilla, formats them, produces lattices, trains the LatticeTM WFST model, and produces predicted transcriptions. See Figure 2. Our evaluation pipeline predicts phoneme transcriptions from audio via Allosaurus only (as a baseline comparison) and uses Epitran to generate ground-truth phone sequences from downloaded transcriptions before computing phone error rate. See section 3.3.

Training Hyperparameters: We trained all models using 5 epochs with a training set of 1650 examples and a test set of 187 (with the exceptions of Persian, which used a test set of 166 and Tulu,

[4]https://www.openfst.org/twiki/bin/view/Contrib/FsmLibrary
[5]https://www.youtube.com/watch?v=mSTSTh8gfRc
[6]https://github.com/apm1467/videocr
[7]https://github.com/yonilevy/noiseclean
[8]https://huggingface.co/datasets/covost2
[9]https://commonvoice.mozilla.org/en

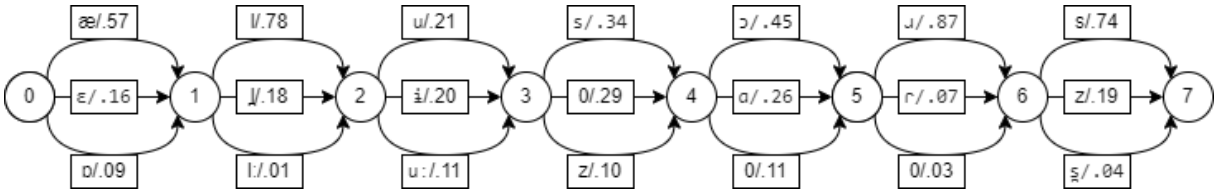

Figure 1: Phone lattice produced by Allosaurus. This is the lattice for an audio recording of the word "allosaurus," with $top\_k = 3$.

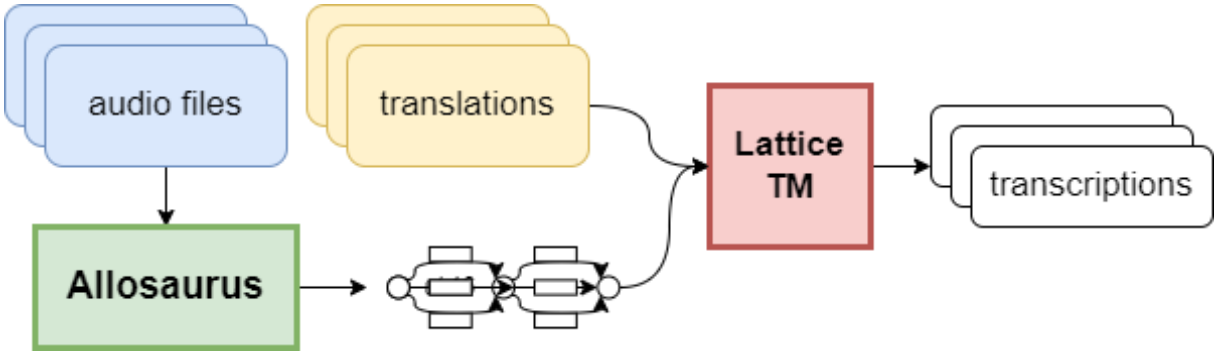

Figure 2: Phone lattice are produced by Allosaurus from speech data. Phone lattices are then passed along with higher resource language translations to LatticeTM to create transcriptions of the audio files.

which used a training set of 1432). We used a default LatticeTM hyperparameters otherwise and a random seed of 4. For lattice production from Allosaurus we used $top\_k = 3$.

3.3 Evaluation

Results evaluation required additional efforts to acquire phoneme transcriptions. The original datasets for Tamil, Arabic, Spanish, Indonesian, and Persian included only speech data corresponding to English translations and orthographic transcriptions, whereas the pipeline's outputs are phoneme IPA transcriptions. Epitran[10], a "tool for transliterating orthographic text as IPA (International Phonetic Alphabet)", was used to create ground truth IPA transcriptions from orthographic transcriptions for evaluation. The pipeline's output was then compared to Epitran's output using the metric of Phoneme Error Rate (PER).

4 Results and Analysis

Experiments show that LatticeTM fails to improve ASR in the language-universal settings we observed. Table 2 shows phoneme error rates for transcriptions from LatticeTM and from Allosaurus alone (as a baseline).

LatticeTM transcriptions in these settings are very poor. Note in figure 3 that LatticeTM transforms Allosaurus transcriptions significantly, and with low accuracy. On the surface, the objective of the WFST model is not terribly difficult. Its task is to choose correctly between the top_k Allosaurus predictions for each frame. This task is simple but non-trivial, since Allosaurus may transcribe a single lexeme with a number of variations, and LatticeTM can resolve these ambiguities.

Lang.	*Allosaurus PER*	*LatticeTM PER*
Tamil	**70.7%**	81.7%
Arabic	**78.4%**	86.8%
Spanish	**48.6%**	78.9%
Indonesian	**64.4%**	81.3%
Persian	**88.7%**	106%

Table 2: Phoneme error rates. **Bold** results are statistically better ($p = .01\%$)

Because accuracy degrades, and $top\_k = 3$, it is not clear if the model is learning at all. This is surprising and may indicate a need for model tuning to new settings, since LatticeTM learned effectively when trained on Spanish word lattices. (See Table 1.) It is also likely that LatticeTM suffers from Allosaurus' poor performance. It may be finding patterns given the aligned translation data, but it may be too difficult to make meaningful predictions when its input lattices are already full of inaccuracies. Explorations into these error causes

[10]https://github.com/dmort27/epitran

are areas of future work.

One observation we made when evaluating the output was that the resulting phone sequence lengths from the model were generally similar lengths to the reference phone sequence lengths or shorter. Our model does not peform many phone insertions, but phone substitutions and deletions. Deletions occur when the model chooses a blank token among Allosaurus' top k (but not top 1) phone options.

4.1 Tulu Performance

Evaluating the resulting phones proved to be somewhat challenging, as there were no ground truths to compare our results with, especially for an extremely low-resource language like Tulu. Furthermore, we were not able to find a native Tulu speaker that could assess the results from the model.

The poor results can largely be attributed to the lack of a proper dataset. Although we were able to extract the subtitles from the videos, oftentimes, the subtitles themselves were slightly inaccurate or were not properly aligned to the audio. There were also subtitles that did not properly get extracted. Additionally, although we attempted to clean the audio clips with noisegating techniques via ffmpeg and sox, we were not able to remove all the background noise from the audio clips, resulting in noisy data; the more we attempted to clean it, the more distorted the audio became. We consulted Dr. Shinji Watanabe for further insight on cleaning audio and he mentioned that these cleaning methods cause special distortions for the audio, which ends up degrading the overall ASR performance. Therefore, we ran our pipeline on other languages with better datasets to evaluate our model's performance.

5 Conclusions and Future Work

We introduce a novel pipeline for ASR via latent lexicon and translation model construction. Though the performance of the model proved to be sub-optimal for all of the languages tested, the reasonable performance of LatticeTM in our preliminary experiment leads us to believe that this pipeline's performance would be more promising if passed in more accurate phone lattices. Inputting phone lattices that consist of high phone error rates leads LatticeTM to not properly learn. To improve the performance of the overall pipeline, methods must be explored to improve the performance of Allosaurus itself. It could be the case that the model would perform more optimally through task-based or language-based fine tuning. This would lead to a stronger correlation between phones and translations.

Furthermore, it is possible that the LatticeTM architecture is insufficient for language universal settings. The system appears to rely on high-quality phone lattices in order to improve performance, and such lattices may be unavailable for many languages (barring significant improvements in the performance of Allosaurus). The WFST translation and lexicon models could be replaced with parametric models such as neural networks. It is possible that improving performance of the lexicon model, typically unobserved, could improve overall ASR pipeline performance.

In addition to altering hyper-parameters and training methods for the initial pipeline, work could be done on data collection for endangered languages, particularly from YouTube movies with subtitles. Research should include cleaning background noise such as music and correctly extracting built-in subtitles from each frame, both of which proved to be very difficult tasks in our experiments. Future experiments for Tulu could also include running the pipeline without attempting cleaning techniques which sometimes augmented the audio beyond recognition.

References

Oliver Adams, Graham Neubig, Trevor Cohn, and Steven Bird. Learning a Translation Model from Word Lattices. In *Proc. Interspeech 2016*, pages 2518–2522, 2016.

Oliver Adams, Graham Neubig, Trevor Cohn, Steven Bird, Quoc Truong Do, and Satoshi Nakamura. Learning a lexicon and translation model from phoneme lattices. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2377–2382, Austin, Texas, November 2016. Association for Computational Linguistics.

P. J. Antony, Hemant B. Raj, B. S. Sahana, Dimple Sonal Alvares, and Aishwarya Raj. Morphological analyzer and generator for tulu language: a novel approach. In *ICACCI '12*, 2012.

Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. An attentional model for speech translation without transcription. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational*

	Spanish	Spanish	Tamil
Reference	elmismotenjapɾoblem askaɾdjakos	peɾonoobtubjeɾonlaɾ enunsiadefalkon	t͡ʃo:lajjile:pu:t̪ t̪at̪anippu:vo:n̪i: t̪a:n
Allosaurus	elnismotenjapɾol̪le maskal̪jakos	peɾanoftubjeanleɾen onfjðet̪alkon	iol̪elepuɾtotal̪ipu ʊrnɪrða
Allosaurus PER	16.67	38.24	61.9
LatticeTM	ɛt̪t̪rakomealfunsææ ɾxelas̪d̪ebanit̪ol̪ asd̪ɾel̪akarβespɾa s̪ɔlse	peɾanotubjɛnnereɔn s̪eɾfafa	olil̪i:ortot̪enituu :ɾiɪ
LatticeTM PER	166.67	55.88	69.05

Figure 3: Example phone sequences produced by Allosaurus and LatticeTM from speech data for various languages.

Linguistics: Human Language Technologies, pages 949–959, San Diego, California, June 2016. Association for Computational Linguistics.

Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anastasopoulos, David R. Mortensen, Graham Neubig, Alan W Black, and Florian Metze. Universal phone recognition with a multilingual allophone system, 2020.

Coskun Mermer and Murat Saraclar. Bayesian word alignment for statistical machine translation. In *ACL (Short Papers)*, pages 182–187, 2011.

Alexis Michaud, Eric Castelli, et al. Towards the automatic processing of yongning na (sino-tibetan): developing a'light'acoustic model of the target language and testing'heavyweight'models from five national languages. In *4th International Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU 2014)*, pages 153–160, 2014.

Graham Neubig, Masato Mimura, Shinsuke Mori, and Tatsuya Kawahara. Bayesian learning of a language model from continuous speech. *IEICE Trans. Inf. Syst.*, 95-D:614–625, 2012.

Matthias Paulik and Alex Waibel. Training speech translation from audio recordings of interpreter-mediated communication. *Comput. Speech Lang.*, 27(2):455–474, February 2013.

Joris Pelemans, Tom Vanallemeersch, Kris Demuynck, Hugo Van hamme, and Patrick Wambacq. Efficient language model adaptation for automatic speech recognition of spoken translations. 09 2015.

B.S. Shivakumar. A wordnet for tulu. 2010.

E. Vidal, F. Casacuberta, L. Rodriguez, J. Civera, and C.D.M. Hinarejos. Computer-assisted translation using speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3):941–951, 2006.

li Zezhong, Hideto Ikeda, and Junichi Fukumoto. Bayesian word alignment and phrase table training for statistical machine translation. *IEICE Transactions on Information and Systems*, E96.D:1536–1543, 07 2013.