EFFECTIVENESS OF MINING AUDIO AND TEXT PAIRS FROM PUBLIC DATA FOR IMPROVING ASR SYSTEMS FOR LOW-RESOURCE LANGUAGES

Kaushal Bhogale\textsuperscript{λψ} Abhigyan Raman\textsuperscript{ψ} Tahir Javed\textsuperscript{λψ} Sumanth Doddapaneni\textsuperscript{λψ} Anoop Kunchukuttan\textsuperscript{ψλ} Pratyush Kumar\textsuperscript{ψ§} Mitesh M. Khapra\textsuperscript{λψ}

\textsuperscript{λ}Indian Institute of Technology, Madras  \textsuperscript{ψ}AI4Bharat  \textsuperscript{§}Microsoft

ABSTRACT

Collecting labelled datasets for speech recognition systems for low-resource languages on a diverse set of domains and speakers is expensive. In this work, we demonstrate an inexpensive and effective alternative by “mining” text and audio pairs for Indian languages from public sources, specifically from the public archives of All India Radio. As a key component, we adapt the Needleman-Wunsch algorithm to align sentences with corresponding audio segments given a long audio and a PDF of its transcript, while being robust to large errors due to OCR, extraneous text, and non-transcribed speech. We thus create Shrutilipi, a dataset which contains over 6,400 hours of labelled audio across 12 Indian languages totalling to 3.3M sentences. We establish the quality of Shrutilipi with 21 human evaluators across the 12 languages. We also establish the diversity of Shrutilipi in terms of represented regions, speakers, and mentioned named entities. Significantly, we show that adding Shrutilipi to the training dataset of ASR systems improves accuracy for both Wav2Vec and Conformer model architectures for 7 languages across benchmarks.

Index Terms—— Alignment, Low-Resource Languages

1. INTRODUCTION

Current state-of-the-art speech recognition systems for high-resource languages often employ end-to-end (E2E) models [1, 2, 3] that require compute-heavy training on large datasets of labelled audio. While the presence of large datasets in high-resource languages helps in reducing the word-error-rate (WER), its absence in low-resource languages increases the performance gap of resulting models, further disadvantaging the adoption of AI models for low-resource languages.

A robust approach to address this gap is to collect labelled datasets for low-resource languages. However, creation of high quality datasets can be expensive given the logistics of collecting data across a diverse set of domains and speakers. Another approach is to utilize unlabelled data using approaches like self-supervised learning or use high-resource languages to boost performance on low-resource languages by transfer learning. This was demonstrated for Indian languages [4] with pretraining on 40 languages and finetuning with only 40 hours of data. However, the WER reported for Indian languages is still much higher than what is achieved with equivalent models for high-resource languages. Cross-lingual transfer from high to low-resource was demonstrated for Indian languages, specifically by creating labelled datasets using transliteration [5]. However, the resultant WER values are still large. Thus, the role of larger datasets is imperative to further reduce the large gap to high-resource languages.

In this paper, we propose an alternative technique to create diverse datasets for low-resource languages by mining audio and text pairs from publicly available sources, specifically focusing on the news archives from All India Radio (AIR). The AIR website\textsuperscript{1} hosts thousands of hours of audio and PDF transcripts for programs. We work with data for 12 Indian languages, containing over 9,700 hours of audio, collectively representing 1.1B number of speakers in the Indian subcontinent with a geographical diversity as shown in Figure 1.

We encounter several irregularities in the AIR data which make mining challenging. The audio bulletins usually contain long intro and outro music segments, short non-transcribed speech (such as speakers introducing themselves, reading titles of the broadcast, or social media handles of AIR), long

\textsuperscript{1}https://newsonair.gov.in/
non-transcribed speech (such as external news clips), background music and code-mixed data. The PDF transcripts also have challenges such as containing proprietary encodings due to legacy issues, custom formats varying across stations, extraneous text such as bulletin and section headers.

Motivated by these challenges, we propose a document-scale alignment technique between a given long audio and its corresponding transcript that is robust to the irregularities discussed. First, for the given audio, we obtain frame-level emissions from an ASR system and collapse them into a character sequence using CTC [2] alignment. Second, we adapt the Needleman-Wunsch [6] algorithm to align the outputs of the ASR system and text extraction. Third, we segment the audio using the sentence boundaries of the aligned text. We retain each pair of segmented text and audio if the alignment score is above a chosen threshold \( \tau \).

We demonstrate the effectiveness of our technique by creating the Shrutilipi dataset with 6,457 hours of data at a threshold of \( \tau = 0.8 \), which is more than 60% of the raw audio. We evaluate the quality of Shrutilipi, with 4,050 pairs of aligned text and audio using the sentence boundaries of the aligned text. We find the ASR system and text extraction. Third, we segment the audio into sentences in \( R \) and collapse repeated characters and remove \{blank\} tokens from the emissions to get the predicted sequence of characters \( P = \{p_1, p_2, ..., p_N\} \). We store the CTC alignments, i.e., the start and end indices of emissions that correspond to \( p_i \) in a sequence \( A = [(\gamma_1, \delta_1), (\gamma_2, \delta_2), ..., (\gamma_j, \delta_j)] \) where \( \gamma_i \) and \( \delta_i \) denote the start and end index respectively.

We then use the Needleman-Wunsch [6] algorithm to align predicted text \( P \) and the reference text \( R \). Given sequences \( R \) and \( P \) of sizes \( N \) and \( N' \) respectively, we find a mapping \( M \), which is a non-decreasing map of every index \( i \) of \( R \), to an index \( M(i) \) of \( P \). This is optimized based on a score matrix \( S \) of size \((N + 1) \times (N' + 1)\), where \( S_{jk} \) denotes the alignment score of characters \( P[j \cdot j] \) and \( R[i \cdot k] \). We score pairs of characters with three values: a Match score when the two characters exactly match, a Mismatch score where the two characters do not match, and a Gap score when the chosen alignment involves one character aligning to a gap in the other sequence. The values chosen for the three scores is application dependent; based on empirical evaluation we penalize mismatch and gap equally.

Given character-level scores, dynamic programming is used to find the mapping \( M = \arg \min_{O} \sum_{i=1}^{N} S_{i,O[i]} \). Once we compute \( M \), we can obtain alignments of sentences in \( R \) to time intervals. The \( i \)th sentence in \( R \) maps to the character range \([\alpha_i, \beta_i] \) which in turn map to the character range in \( P \): \([M(\alpha_i), M(\beta_i)] \), which in turn map to the character range in \( E \): \([A(M(\alpha_i)), A(M(\beta_i))] \), which map directly to indices in the input audio \( X \).

As the Needleman-Wunsch algorithm does global alignment, it is possible that it misaligns certain segments to optimize the scores for other segments. Hence, we need a filtering mechanism for extracting high-quality audio-text sentence pairs. To address this, we define an alignment score given by the Levenshtein distance similarity ratio between the mined reference sentence \( r \) and predicted sentence \( p \), defined as \( \Delta = 1 - \frac{LD(r, p)}{|r| + |p|} \). We store the CTC alignments in \( \Delta \) for the three scores is application dependent; based on empirical evaluation we penalize mismatch and gap equally.
Table 1: Statistics of Shrutilipi dataset (# M.W.=Mean length of Sentences (in words))

| Lang. | bn | gu | hi | kn | ml | mr | or | pa | sa | ta | te | ur | Total |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| # Hrs. | 0.44K | 0.46K | 1.62K | 0.46K | 0.36K | 1.02K | 0.60K | 0.09K | 0.02K | 0.79K | 0.39K | 0.19K | 6.46K |
| # Sents. | 0.23M | 0.22M | 0.79M | 0.22M | 0.26M | 0.49M | 0.30M | 0.04M | 0.01M | 0.41M | 0.22M | 0.10M | 3.30M |
| # M.W. | 16 | 17 | 21 | 13 | 9 | 16 | 15 | 26 | 11 | 13 | 13 | 22 | 14.8 |
| Yield(%) | 69 | 67 | 69 | 71 | 42 | 79 | 78 | 78 | 42 | 71 | 47 | 63 | 67 |

3. SHRUTILIPI DATASET

In this section, we discuss applying the mining procedure to the AIR archive to create Shrutilipi in 12 languages. We use IndicWav2Vec [4] models trained on Kathbath [15] to perform ASR Prediction using CTC. We set the match, mismatch, and gap scores of the Needleman-Wunsch alignment to $+2$, $-1$, and $-1$, respectively, based on experiments to maximize the yield of aligned data. We chose the value of $\tau = 0.8$, thereby retaining aligned pairs with some errors in anticipation that training an ASR system would benefit from a larger volume of labelled pairs. We confirm that this is indeed the case in our experiments. We use Google’s Document AI OCR to extract text from the PDF documents. By applying the method to the AIR archive we extract 6,457 hours of data across 12 languages as detailed in Table 1, a yield of 67% of all audio available. The data corresponds to 3.3M utterances across 12 languages.

In the following, we evaluate the effectiveness of our mining technique, by analyzing Shrutilipi along three axes: quality, diversity, and effectiveness for downstream ASR.

3.1. Is the data of good quality?

We perform a human evaluation of Shrutilipi with data sampled across languages, regions, and alignment quality. The task is to check the quality of a mined audio-text pairs. First, evaluators were shown the text and were asked if there were any mistakes in the text. This is to capture potential errors from the text processing pipeline. Then, evaluators listened to the audio and were asked if the audio aligns with the text. If they answered ‘No’, they were asked to localize the error to (a) Start, (b) In between, and (c) End of sentence. We sample data uniformly across the 45 radio stations across all languages and three intervals of the alignment score $\{[0.8 - 0.9], [0.9 - 0.95], [0.95, 1]\}$. For each combination of the 45 stations and 3 score intervals, we uniformly sample 30 audio-text pairs, creating an annotation dataset with 4,050 items.

In Figure 2(b), we plot the fractions of responses to different questions against different values of alignment score. We make three observations. First, as expected, the fraction of errors reduces as alignment scores increase, with a marked reduction around the value of 0.95. Second, a large fraction of the errors (in the range $[0.8, 0.9]$) are due to errors in the original text, indicating the need for more accurate OCR and document understanding for Indian languages. Third, when localizing the error in alignment, a majority of the errors seem to be at the start or end of the audio segments. Only a smaller fraction of errors are due to alignment issues within the audio segment, which incidentally do not show a strong dependence on the alignment score. We hypothesize training methodologies for E2E ASR systems would be forgiving of such errors.

3.2. Is the data diverse?

A key metric for labelled audio datasets is the diversity of speaker and content representation [18]. We compute metrics of diversity and compare against the MUCS dataset [7].

To quantify speaker diversity, we build an speaker verification model to compute speaker embeddings. Specifically, we use the X-Vector model [19], trained on Kathbath [15]. We randomly sample 10K pairs of audio segments from the Hindi train sets of Shrutilipi and MUCS [7], compute the cosine similarity of these pairs, and plot its distribution in Figure 2(c). The distribution for Shrutilipi has a very large fraction of smaller similarity scores, indicating larger diversity.

To quantify diversity of source-native Named Entities (NEs), we count the number of occurrences of NEs in the Hindi datasets of Shrutilipi and MUCS [7]. We translate the sentences to English using IndicTrans [20], and then use Spacy’s Entity Recognizer to obtain NEs. We observe that MUCS and Shrutilipi datasets contain 766 and 222K unique NEs respectively, i.e., Shrutilipi provides a $290\times$ increase. Further, the fraction of words that are NEs is also much larger in Shrutilipi across entity types (Figure 2(c)).

3.3. Is it effective on downstream ASR?

We evaluate the effectiveness of Shrutilipi as a training dataset for ASR systems. We consider both a large model - Wav2Vec [8], and an efficient model - Conformer [3].

For training Wav2Vec models, we use the LARGE model (317M parameters), and finetune for 120K steps, by initializing the model from the pretraining checkpoint. We use 6-gram KenLM [21] language models trained on the IndicCorp corpus [22], and evaluate models using beam search decoding.
Fig. 2: (a) Shrutilipi significantly increases the amount of labelled ASR data in comparison with existing publicly available datasets. (b) Fraction of annotation errors in Shrutilipi reduces as alignment scores (Δ) increase. (c) Shrutilipi has more diversity in speakers and named entities compared to MUCS [7].

| MUCS Blind Set | | | | | | | Avg. |
|----------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| E              | 17.9             | 12.0             | 13.6             | 23.3             | 20.5             | 16.4             | 17.3             |
| E+S            | 12.8             | 11.1             | 11.4             | 23.0             | 20.7             | 13.8             | 15.5             |

| Kathbath Test Unknown | | | | | | | Avg. |
|-----------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| E                     | 14.4             | 15.0             | 14.7             | 25.6             | 31.5             | 24.1             | 22.3             | 21.1             |
| E+S                   | 13.4             | 9.5              | 9.6              | 15.7             | 21.5             | 19.7             | 17.7             | 15.3             |

Table 2: Results for Wav2Vec models on the Test Unknown of Kathbath (E = Existing; S = Shrutilipi)

| Benchmarks | KB-K | KB-U | EKST | CV6 | CV7 | CV8 | CV9 | Avg. |
|------------|------|------|------|-----|-----|-----|-----|------|
| M_W2V      | 14.1 | 14.7 | 22.7 | 31.3 | 26.7 | 27.5 | 25.4 | 23.6 |
| M + S_W2V  | 9.4  | 9.6  | 19.7 | 15.0 | 15.0 | 13.4 | 13.9 | 13.7 |

|                      |                      |                      |                      |                      |                      |                      |
|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| M + S_{r=0.95}      | 9.8                  | 10.2                 | 20.4                 | 16.4                 | 14.2                 | 14.9                 | 14.7               | 14.4              |

|                      |                      |                      |                      |                      |                      |                      |
|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| M_{Hard}             | 19.2                 | 17.9                 | 26.5                 | 22.6                 | 24.4                 | 25.6                 | 25.8               | 23.2              |
| M + S_{Hard}         | 12.1                 | 12.6                 | 23.4                 | 18.7                 | 17.4                 | 18.6                 | 18.4               | 17.3              |

|                      |                      |                      |                      |                      |                      |                      |
|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| M_{Conf.}            | 17.2                 | 17.7                 | 25.4                 | 20.9                 | 21.4                 | 22.9                 | 22.8               | 21.2              |
| M + S_{Conf.}        | 15.2                 | 14.9                 | 23.9                 | 19.3                 | 19.1                 | 20.0                 | 19.9               | 18.9              |

Table 3: Results on Hindi Benchmarks for Wav2Vec and Conformer models trained on MUCS and Shrutilipi datasets (M = MUCS; S = Shrutilipi; W2V = Wav2Vec; Hard = hard benchmark; Conf = Conformer; KB = Kathbath; K = Known; U = Unknown; EKST = EkStep; CV = CommonVoice)

with a beam size of 128. For Conformer models, we use the Medium model consisting of 30.5M parameters. We replace the LSTM decoder with a CTC decoder and instead increase decoder layers to 18, and evaluate using greedy decoding.

We evaluate performance of Wav2Vec [8] models on the blind set of MUCS [7] and Test Unknown set of Kathbath [15], as shown in Table 2. The model for Bengali was trained on the OpenSLR [23] train set, and the MUCS train set for others, denoted by E (Existing). For the MUCS blind set, the average WER drops from 17.3 to 15.5. For Kathbath too, we see a large improvement of 5.8 WER on average.

Hindi has 7 benchmarks for which we report results separately in Table 3. We evaluate Wav2Vec models trained on the MUCS [7] train set and MUCS+Shrutilipi. We see a consistent improvement in WER across all the 7 benchmarks, with an average improvement of 5.3 WER. We also see that the model trained on Shrutilipi (τ = 0.8) performs better than Shrutilipi (τ = 0.95) (row 2 vs 3 of Table 3), indicating that data volume outweighs some alignment errors localized at the edges of mined utterances. We train the Conformer model on MUCS train set and MUCS+Shrutilipi, and evaluate on the Hindi Benchmarks. Again, we see consistent improvement in WER across all benchmarks, wherein the Average WER improves from 21.2 to 18.9 (row 6 vs 7 of Table 3).

To evaluate if addition of Shrutilipi to the training set makes the models more robust to noise, we create a hard ASR benchmark for Hindi by adding background noise of various types to the audio files of the Hindi Benchmarks. We use ESC dataset [24], which consists of 2,000 short clips of background noise from 5 different categories. For each audio, we randomly pick a background clip and add it to the audio signal with a random Signal-to-Noise Ratio (SNR) value between 3 dB and 30 dB. As seen in rows 4 and 5 of Table 3, WERs of all models increase with this hard benchmark. Moreover, models trained with Shrutilipi have smaller increases (3.8 WER) as compared to others (4.4 WER).

Thus, we establish that Shrutilipi improves performance of ASR systems across languages and model architectures.

4. CONCLUSION

We demonstrate creation of datasets for low-resource languages by mining data from the archives of AIR. Our technique for document-scale alignment is robust to large irregularities both in the audio and available transcript PDFs. The resultant Shrutilipi dataset, consists of 6,457 hours of audio across 12 Indian languages, significantly adding to available resources in these languages. We showed that the data is of high quality with human evaluation, is diverse in terms of speakers and domains, and is effective in improving accuracy of ASR systems. We open-source our dataset and models2. We hope that this template can be replicated with other sources of data and languages to bridge the widening gap between ASR systems for high and low-resource languages.

2https://ai4bharat.org/shrutilipi

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
5. REFERENCES

[1] Jinyu Li et al., “Recent advances in end-to-end automatic speech recognition,” APSIPA, vol. 11, no. 1, 2022.

[2] Alex Graves, “Connectionist temporal classification,” in Supervised sequence labelling with recurrent neural networks, pp. 61–93. Springer, 2012.

[3] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., “Conformer: Convolution-enhanced transformer for speech recognition,” arXiv preprint arXiv:2005.08100, 2020.

[4] Tahir Javed et al., “Towards building asr systems for the next billion users,” in AAAI, 2022, vol. 36, pp. 10813–10821.

[5] Shreya Khare, Ashish R Mittal, Anuj Diwan, Sunita Sarawagi, Preethi Jyothi, and Samarth Bharadwaj, “Low resource asr: The surprising effectiveness of high-resource transrating,” in Interspeech, 2021, pp. 1529–1533.

[6] Saul B Needleman and Christian D Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” Journal of molecular biology, vol. 48, no. 3, pp. 443–453, 1970.

[7] Anuj Diwan et al., “Multilingual and code-switching asr challenges for low resource indian languages,” arXiv preprint arXiv:2104.00235, 2021.

[8] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” Neurips, vol. 33, pp. 12449–12460, 2020.

[9] Andrej Ljolje and MD Riley, “Automatic segmentation and labeling of speech,” in Acoustics, Speech, and Signal Processing, IEEE International Conference on. IEEE Computer Society, 1991, pp. 473–476.

[10] Florian Schiel, “Automatic phonetic transcription of non-promted speech,” 1999.

[11] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger, “Montreal forced aligner: Trainable text-speech alignment using kaldi.,” in Interspeech, 2017, vol. 2017, pp. 498–502.

[12] Kishore Praballad, Automatic building of synthetic voices from audio books, Ph.D. thesis, Carnegie Mellon University, 2010.

[13] Elizabeth E Shriberg, “Phonetic consequences of speech disfluency,” Tech. Rep., SRI INTERNATIONAL MENLO PARK CA, 1999.

[14] Athanasios Katsamanis, Matthew Black, Panayiotis G Georgiou, Louis Goldstein, and Shrikanth Narayanan, “Sailalign: Robust long speech-text alignment,” in Proc. of workshop on new tools and methods for very-large scale phonetics research, 2011.

[15] Tahir Javed et al., “Indicsuperb: A speech processing universal performance benchmark for indi languages,” CoRR, vol. abs/2208.11761, 2022.

[16] GIZ, “A study on open voice data in indi languages,” https://toolkit-digitalisierung.de/app/uploads/2021/02/Study-on-Open-Voice-Data-in-Indian-Languages.pdf, 2021, Accessed: 2022−10−08.

[17] Tahir Javed et al., “Indicsuperb: A speech processing universal performance benchmark for indi languages,” 2022.

[18] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber, “Common voice: A massively-mulilingual speech corpus,” arXiv preprint arXiv:1912.06670, 2019.

[19] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in ICASSP. IEEE, 2018, pp. 5329–5333.

[20] Gowtham Ramesh, Sumanth Doddapaneni, et al., “Samantar: The largest publicly available parallel corpora collection for 11 indic languages,” TACL, vol. 10, pp. 145–162, 2022.

[21] Kenneth Heafield, “KenLM: Faster and smaller language model queries,” in Proceedings of the Sixth Workshop on Statistical Machine Translation, Edinburgh, Scotland, July 2011, pp. 187–197, Association for Computational Linguistics.

[22] Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar, “Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages,” in EMNLP, 2020, pp. 4948–4961.

[23] Vishwas M Shetty and Srinivasan Umesh, “Exploring the use of common label set to improve speech recognition of low resource indian languages,” in ICASSP. IEEE, 2021, pp. 7228–7232.

[24] Karol J Piczak, “Esc: Dataset for environmental sound classification,” in Proceedings of the 23rd ACM international conference on Multimedia, 2015, pp. 1015–1018.