Vāgyojaka : An Annotation and Post-Editing Tool for Automatic Speech Recognition

Rishabh Kumar[1*], *Devaraja Adiga*[1*], *Mayank Kothari*[1], *Jatin Dalal*[2], *Ganesh Ramakrishnan*[1], *Preethi Jyothi*[1]

[1]Indian Institute of Technology, Bombay, [2]Indian Institute of Information Technology, Una

krrishabh@cse.iitb.ac.in, pdadiga@iitb.ac.in, maykat2017@gmail.com, jd.jatindalal@gmail.com, ganesh@cse.iitb.ac.in, pjyothi@cse.iitb.ac.in

Abstract

Vāgyojaka is an open-source post-editing and annotation tool for automatic speech recognition (ASR) that aims to reduce the human effort required to correct the ASR results. We adopt a dictionary-based lookup method to highlight the incorrect words in the ASR transcript and give suggestions by generating the closest valid words. For curating the speech corpus, we provide a rich list of tagset that captures various spoken audio features. Further, we conducted a user study to evaluate the effectiveness of our tool and observed that post-editing requires 1/3 lesser time than editing without using our tool. The user study can be found on our website [1].

Index Terms: Automatic speech recognition, post-editing of ASR transcript, speech corpus annotation

1. Introduction

With the advent of deep learning, automatic speech recognition (ASR) has improved significantly in recent years. However, the results are not fully accurate[1, 2] and are often post-corrected by the human editors. Several attempts have been made to reduce the post-editing effort by developing assistive tools such as GECKO [3]. However, it lacks the facility to ingest video files. Another tool, Beey [4] provides several post-processing features which is not freely available. In this work, we present Vāgyojaka, an open-source ASR post-editing tool with features such as multi-language support, video transcript alignment, error detection, *etc.* Our tool is inspired from Optical Character Recognition (OCR) post-processing and translation tool [5, 6].

Many low-resource languages require human annotation to build labeled corpora for training. In many cases, when the domain of the test sample changes, the quality of the ASR system significantly deteriorates. For example, an ASR system trained using recordings of 56-hour readings from various Sanskrit books gives 21.94% of word error rate (WER) while the WER for a live lecture is 51.52% [1].

2. Vāgyojaka : Annotation and Post-Editing Tool

Vāgyojaka is a standalone offline application built using QT-creator. It provides an interactive user interface to edit ASR predictions while focusing on capturing various audio features in a multi-lingual environment. Below we describe the salient features of our tool.

[*]Equal Contribution

[1]https://www.cse.iitb.ac.in/~asr/VAgyojaka

2.1. Features

1. User Interface: As shown in the figure 1, the uploaded video or audio file is played on the left side of the tool while the corresponding ASR output appears on the right half. The Editor tab at the top shows many shortcut keys with the associated tasks relevant to the editor. Each line begins with the speaker's name, followed by the ASR transcript and the timestamp at which the current line's utterance in the media ends. The line matching the current time frame in the player is presented with a green-coloured font. On the bottom side of the text editor, a separate editor for augmenting information at the word level may be toggled on or off.

2. Video-transcript alignment using timestamps: A user can watch the video (or listen to the audio) while referring to the ASR output for that time frame using the tool. It highlights the corresponding transcript section based on the timestamp.

3. Speaker database: A user can add a new speaker and save it to a speaker database. The user can either change the speaker's name for all the utterances of that particular speaker or a particular sentence. Furthermore, annotators can edit speaker-specific transcripts by listening to one speaker while skipping other speakers in the middle.

4. Highlight the sentence: While the video/audio is playing, the transcript sentence matching that time frame is automatically highlighted, as shown in 1. This allows the user to locate and edit the line quickly.

5. Time propagation dialogue: The timestamp changes are in an absolute number, wherein common intervals of time can be added/subtracted for a range of the sentences or a particular sentence based on the user's choice.

6. Fix timestamps: A user can change/insert the current timestamp of the player in any line/word using the keyboard shortcut to match the sentence boundary.

7. Multi-language support: Multiple languages are supported by the tool, which includes text completion and suggestions in the editor. To make relevant recommendations, the user may also add a domain-specific dictionary.

8. Error Detection: The tool leverages a fixed vocabulary to detect incorrect words in the ASR transcript. These incorrect words are highlighted using a red underline. Currently, we are supporting English and five Indic languages, viz. Hindi, Sanskrit, Tamil, Gujarati and Telugu.

9. Log: The Vāgyojaka logs all the changes made in the text editor with position and characters, which can be used to analyze annotation and further improve the ASR system and use it as a particular research problem.

10. Capturing audio features: There are multiple features exhibited by audio which can be classified into the acoustic level and language level features. Acoustic features are those such

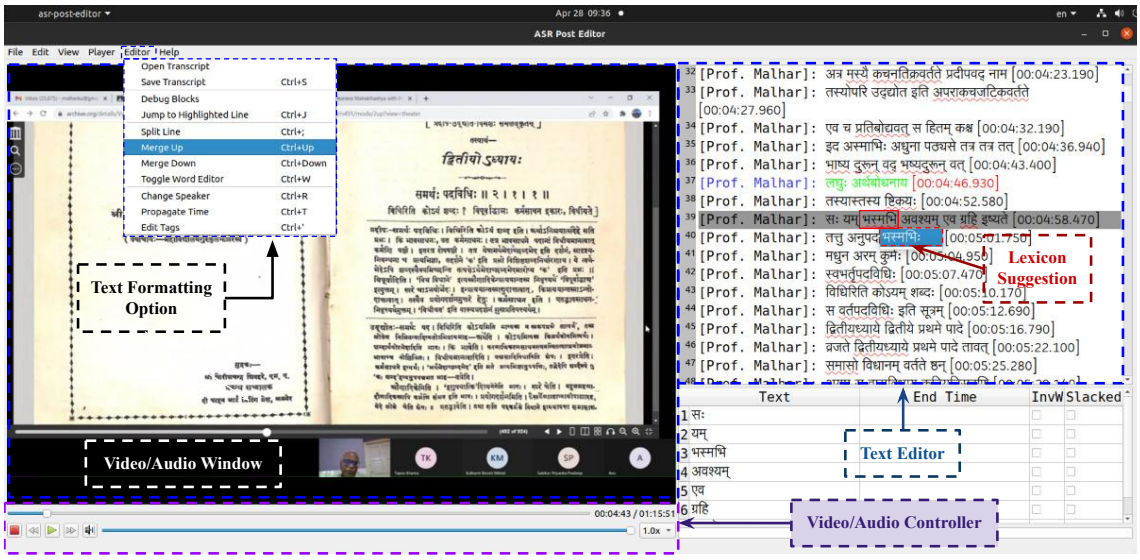

Figure 1: *Screenshot of the ASR Post-Editing tool*

as noise level, influence of the L1 language when the speaker is rendering L2 or L3 language, slurred speech, etc., while the language features are like the language being spoken in a multi-language environment, validness of a sentence or word, etc. These features are essential to building a high-quality ASR system for a low-resource language and code-switching. To improve quality data curation during the annotation process, the following tags can be added at a sentence or word level as required.

- <Noisy> If the audio is noisy.
- <MultipleSp> If multiple speakers speaking at a time.
- <Language>(a Dropdown List) If audio belongs to different language.
- <L1influence> Influence of L1 language.
- <InvalidS> If sentence is not a valid sentence.
- <InvalidW>(word level tag) If a word is not valid.
- <Slurred> Poorly pronounced words or sentences.
- <NativeLanguage> The speaker's native language (a Dropdown List preceded by NL, such as NL Hindi)

2.2. User Study

We measure the time spent correcting the document from the Vāgyojaka tool compared to post-editing ASR output with a simple word editor. Our study consists of a Sanskrit lecture video and Indo-English accent lecture audio containing 16 sentences having 128 words and 10 sentences having 192 words, respectively. Five Sanskrit and English language experts familiar with the Vāgyojaka tool are involved in the experiment. We distribute the tasks in a round-robin manner to avoid transcription bias. We observed that volunteers who used Vāgyojaka tool achieved an average transcription speedup by a factor of 3 in comparison to post-editing using a simple text editor.

3. Conclusion and Future Work

We introduce Vāgyojaka, an annotation and post-editing tool for speech recognition systems. The interface is designed to reduce the effort and time required to post-edit the transcripts. We have provided a rich tag set which captures various features of an uttered audio to create a gold-quality speech dataset for low-resource languages, which will be helpful in tasks like code-switching, improving ASR results for different subdomains, etc. While researchers from the speech domain appreciated the audio features, logs, etc., collected during the annotation process, annotators reported improvements in the speed of the data curation process.

We are working on adding transliteration functionality so that users may easily input text in languages other than English and incorporate speaker diarization. We will further improve the suggestion quality by leveraging N-best results from an ASR system as an external source. We also plan to develop a web-based alternative to this tool.

Acknowledgement: We would like to thank Ashish Mittal, IBM Research India for his insightful suggestions.

4. References

[1] D. Adiga, R. Kumar, A. Krishna, P. Jyothi, G. Ramakrishnan, and P. Goyal, "Automatic speech recognition in sanskrit: A new speech corpus and modelling insights," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, 2021.

[2] A. Diwan, R. Vaideeswaran, S. Shah, A. Singh, S. Raghavan, S. Khare, V. Unni, S. Vyas, A. Rajpuria, C. Yarra *et al.*, "Multilingual and code-switching asr challenges for low resource indian languages," *Proceedings of Interspeech 2021*, 2021.

[3] G. Levy, R. Sitman, I. Amir, E. Golshtein, R. Mochary, E. Reshef, R. Reichart, and O. Allouche, "Gecko-a tool for effective annotation of human conversations." in *INTERSPEECH*.

[4] L. Weingartová, V. Volná, and E. Balejová, "Beey: More than a speech-to-text editor," *Proc. INTERSPEECH 2021*.

[5] R. Saluja, D. Adiga, G. Ramakrishnan, P. Chaudhuri, and M. Carman, "A framework for document specific error detection and corrections in indic ocr," in *14th ICDAR*, vol. 4.

[6] A. Maheshwari, A. Ravindran, V. Subramanian, A. Jalan, and G. Ramakrishnan, "Udaan-machine learning based post-editing tool for document translation," *preprint arXiv:2203.01644*, 2022.