Interspeech 2018
2-6 September 2018, Hyderabad

CACTAS - Collaborative Audio Categorization and Transcription for ASR Systems

Mithul Mathivanan, Kinnera Saranu, Abhishek Pandey, Jithendra Vepa

Samsung R&D Institute, India - Bangalore

mithul.m@samsung.com, kinnera.sar@samsung.com, abhi3.pandey@samsung.com, jithendra.v@samsung.com

Abstract

We present a web based tool that allows collaborative analysis and/or transcription of audios with respect to Automatic Speech Recognition (ASR) systems. The tool presents a webpage consisting of audios and their corresponding references and hypotheses obtained offline. Several other information and features are provided that allow the audios to be categorized and references to be corrected efficiently in a collaborative way almost 10 times faster, without the need for prior knowledge on speech or ASR systems. The analysis can later be summarized and acted upon to improve or triage the ASR system.

Index Terms: speech recognition, human-computer interaction, collaborative analysis, collaborative correction

1. Introduction

The advent of deep learning based approaches paved the way for ASR technology [1] into commercial applications. These technologies and in particular Voice Assistants, are expected to understand users utterances flawlessly with new domains and categories being added everyday. The ASR Engine of such Voice Assistants must be able to adapt and continuously add support for such new domains quickly.

The end-to-end process of adding a new domain is quite complex and time-consuming considering the difficulty in getting transcribed audio pertaining to the domain, cleaning up the audio, testing ASR system with new data, categorizing and analyzing failures and improving the system performance while ensuring minimum regression in previously supported domains. It is further complicated by the need to work with different kinds of people. While some may be engineers, some others may be pure linguists who do not have much knowledge about the ASR system. Organizations often resort to crowd-sourcing for certain tasks such as voice data collection, transcription, error correction, error analysis and so on where people can be from any background. People may even work from different countries and in different time zones. Managing the whole process and ensuring collaboration [2] and co-operation is key to achieve efficiency. Our system has been designed with these goals in mind. Our tool has several features that we describe in the next sections that help in achieving these goals.

2. System architecture

Our system is built using Ruby on Rails and it is interfaced with ASR servers to run models as and when required. The tool triggers ASR server to get the ASR output on a set of audio files. As shown in Figure 1, it generates a webpage containing the audios (playable using the browser through HTML5), transcription/reference, ASR output, Category, Comment, perplexity of the sentences and Inverse Text Normalization (ITN) output. The

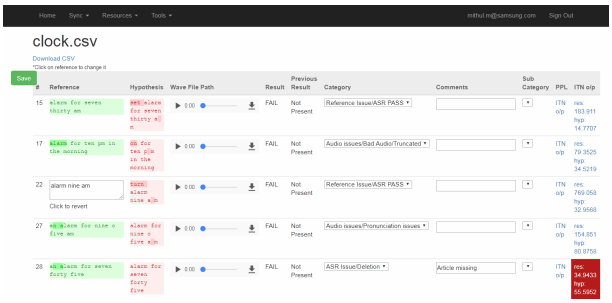


Figure 1: *Screenshot of analysis webpage*

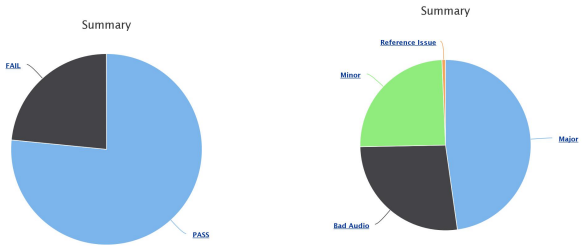
webpage captures all the required information holistically at one place in an easily accessible manner as shown in Figure 1. The information is also presented in a very intuitive manner. Collaboration is enabled by providing a mechanism to split the rows/audios into batches and providing the batch links to several users, who can concurrently work on the assigned page. Users can correct and thereby transcribe the given reference in case of mismatch. Presentation of all of this information at one place rather than having to search multiple sources on an average reduces the time per audio by 50%, with additional features (discussed later in this paper) further reducing the time required. In the following sub-sections, we describe how the tool can be used for different activities.

2.1. Generating ASR Output

The tool indexes the models present on the server and allow users to select the models and set of audios or test cases (TCs) of their choice. This then sends the request to the server to start an ASR Engine with the model specified and obtain the ASR outputs for the audios specified. The results are then retrieved by the tool and compared with specified references to generate the accuracies and a CSV file of the same.

2.2. Transcribing the Audios

In case of transcription the references are left blank since the intended reference of the audio is not known. Users can listen to the audios and transcibe the audio in the page. The tool then reads the CSV file and allows the users to enter the transcription as a reference and save the CSV. The presence of the ASR outputs simplifies the process as if provides a hint of what the utterance is or could be. Shortcuts are provided to copy the ASR output as is to the reference column.



(a) *Broad level categories* (b) *Detailed categorization of Bad Audios*

Figure 2: *Pie chart showing categorization of audios*

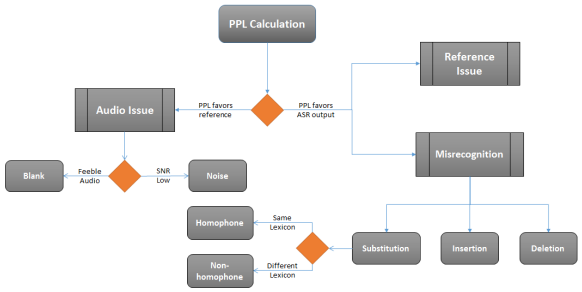


Figure 3: *Flowchart describing simplicity of categorizing ASR outputs*

2.3. Analyzing the ASR output

In case of analysis where the reference is already provided, users are presented with predefined hierarchical categories. Once the first level of category is selected a further detailed level of subcategories is provided for further classification of issues. Users can view the reference, ASR output and listen to the audio and decide the category. The system provides several features to speed up categorization as shown in the following sections and summarized in Table 1.

2.3.1. *Audio features*

- Waveform of the audio allows viewing blank or feeble audios. (Saves 70% time)
- SNR or segmental SNR can be used to identify noisy audios. (Saves 50% time)
- Audio segment pertaining to a specific word of an ASR output can be played for pronunciation/misrecognition analysis. (Saves 20% time)

2.3.2. *Misrecognition features*

- Word level perplexity can give information on which sentence the LM favors. (Saves 50% time)
- Lexicon pronunciations of specific words can be obtained to verify model pronunciation and homophone analysis. (Saves 20% time)
- Word2Vec variations[3] of reference text can be generated with perplexity to verify LM quality on similar sentence structure with variations. (Saves 20% time)

2.3.3. *Reference related features*

- The ASR output can be directly copied in case the reference is wrong. (Can also be used for transcriptions) (Saves 20% time)

Table 1: *Categorization using features*

Broad Category	Sub-category	Feature
Audio Issues	Noise	SNR
	Blank	Flat waveform
	Pronunciation	Play segment
Misrecognition	Homophone substitution	Lexicon pronunciation
	Other Substitution	Word2Vec
	Insertion & Deletion	Perplexity

2.4. Working collaboratively

Since the system is web based it can be split into batches based on number of audios in a page. Multiple users can concurrently work on the system to complete the analysis/transcription faster. The data from multiple users is seamlessly saved without need for manually combining the work.

2.5. Summarizing the analysis

All of the analysis and transcription can be saved as a single xlsx workbook with multiple worksheets or as multiple CSVs those can be viewed separately for further analysis and extract the corrected references or transcriptions. The xlsx file also provides a summary table of how many audios from each domain were classified into what category or subcategory to get an overview of the models performance and improvement areas. The summary of the analyzed categories are also visible on a broad level as shown in Figure 2a and upon selecting the relevant section a drilldown of that category can be viewed as shown in Figure 2b.

A summary of word misrecognition and corresponding replacements with count is given to view a quick summary of substitutions.

3. Conclusions

We have developed a tool that allows users with no prior knowledge on speech to work collaboratively to either transcribe or analyze a given set of audios with minimal training as shown in Figure 3. Our subjective analysis shows the time spent has reduced to about $1/10^{th}$ of the time taken previously, thereby increasing efficiency. Quick and easy summarizing also allows developers to view issues quickly and act upon them easily.

4. References

[1] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[2] P. Bell, J. Fainberg, C. Lai, and M. Sinclair, "A system for real time collaborative transcription correction," *Proc. Interspeech 2017*, pp. 817–818, 2017.

[3] M. Rei and T. Briscoe, "Looking for hyponyms in vector space," in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, 2014, pp. 68–77.