Outcomes of Speech to Speech Translation for Broadcast Speeches and Crowd Source Based Speech Data Collection Pilot Projects

Anil Kumar Vuppala(✉), Prakash Yalla, Ganesh S. Mirishkar, and Vishnu Vidyadhara Raju V

Speech Processing Laboratory, Language Technologies Research Centre, Kohli Center on Intelligent Systems, International Institute of Information Technology, Hyderabad, India
{anil.vuppala,prakash.yalla}@iiit.ac.in,
{mirishkar.ganesh,vishnu.raju}@research.iiit.ac.in

Abstract. Speech-to-Speech Machine Translation (SSMT) applications and services use a three-step process. Speech recognition is the first step to obtain transcriptions. This is followed by text-to-text language translation and, finally, synthesis into text-speech. As data availability and computing power improved, these individual steps evolved. However, despite significant progress, there is always the error of the first stage in terms of speech recognition, accent, etc. Having traversed the speech recognition stage, the error becomes more prevalent and decreases very often. This chapter presents a complete pipeline for transferring speaker intent in SSMT involving humans in the loop. Initially, the SSMT pipeline has been discussed and analyzed for broadcast speeches and talks on a few sessions of Mann Ki Baat, where the source language is in Hindi, and the target language is in English and Telugu. To perform this task, industry-grade APIs from Google, Microsoft, CDAC, and IITM has been used for benchmarking. Later challenges faced while building the pipeline are discussed, and potential solutions have been introduced. Later this chapter introduces a framework developed to collect a crowd-sourced speech database for the speech recognition task.

Keywords: Speech recognition · Machine translation · Speech synthesis · Crowd-source database

Supported by Technology Development for Indian Languages (TDIL), Ministry of Electronics and Information Technology (MeitY), Government of India, for allowing us to work on the "Speech to Speech Translation & performance measurement platform for Broadcast Speeches and Talks" pilot project and "Crowd Sourced Large Speech Data Sets To Enable Indian Language Speech - Speech Solutions".

S. N. Srirama et al. (Eds.): BDA 2021, LNCS 13147, pp. 341–354, 2021.
https://doi.org/10.1007/978-3-030-93620-4_24

1 Introduction

Nowadays, information processing systems have become an integral part of human life. These systems generally take information in one form and process (transform) it into another form. Human-computer interaction (HCI) is predominantly used by considering speech as a source of input as speech signal is a unique and natural way of communication among human beings. Due to the advancement in technology, few researchers were curious about the mechanisms involved in the mechanical realization of human speech abilities, to the desire to automate the simple tasks that are inherently required for HCI. So the motive of the HCI is to listen to human speech and carry out their commands accordingly.

The magnitude of innovations being observed in the fast pace of the deep-learning era made real-time communication possible with a ray of light. By leveraging it, these days, portable devices are being enabled with voice or text-based services as it helps in connecting the global market without any hassle, such as removing the language barriers (wherein speech from one language is taken as an input and transformed into other languages). The dream to have an automatic speech-to-speech machine translation (SSMT) system comes into action if trust exists in deep learning, cognitive computing, and big data models. As of now, most of the SSMT systems [1–6] are compromising with transcription and translation quality. On this grounds, most of the research groups and industries have drawn their interest to mitigate the issue for the smooth functioning of the SSMT pipeline. The SSMT system could be used more broadly in live streaming, customer service management, a wherein person-to-person conversation is needed. These applications will bridge the gap between one language to another, and it would also help in a multilingual society like India.

In building the SSMT system, there involve multiple challenges in all the three blocks which are namely,

- **Automatic Speech Recognition**
  - It is a process of transforming a speech signal to its corresponding text.
- **Machine Translation**
  - It is process in which a text is translated from one natural language to another.
- **Speech Synthesis (text-to-speech)**
  - It converts a given text into speech signal.

From the above mention three blocks, speech synthesis has a decent amount of maturity in the commercial space across the languages. Machine translation (MT) and Speech recognition is still an unsolved problem due to various factors. Coming to ASR, it is mainly due to limited vocabulary size, speaking rate, environmental variations (external noises), different dialects etc.

In this work we have demonstrated the capability of technologies for performing speech to speech translation in Indian context in an industry grade technology stack and capture performance benchmarks to enable industry application grade technology development. Speech to speech translation is performed

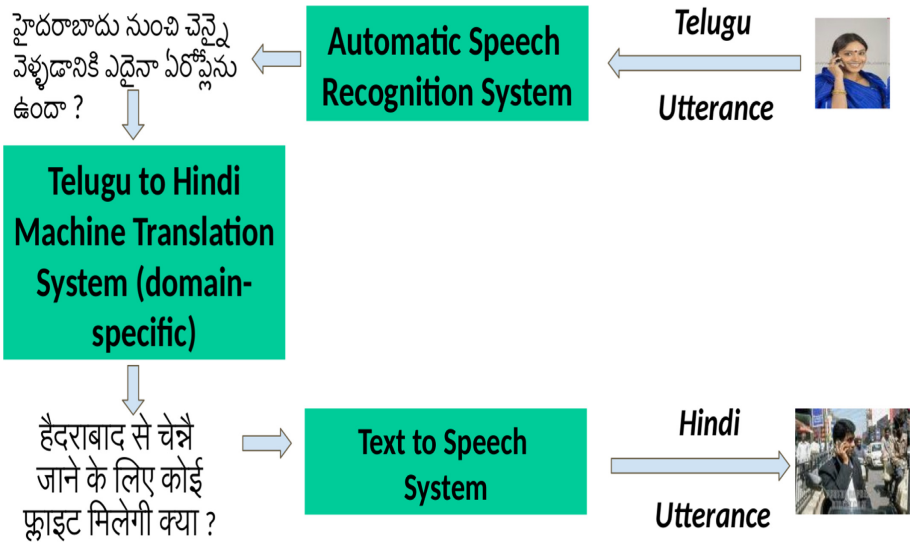

Fig. 1. High-level block schematic of speech to speech machine translation system

using speech recognition, machine translation and text to speech. The typical block diagram for SSMT system is shown in Fig. 1. It is a waterfall architecture, where ASR, MT, and SS systems are joined together to form an SSMT system [7]. The output obtained from an ASR system is given as input to an MT system, and the output obtained from the MT system is given as an input to a SS system. Given accurate ASR and accurate MT systems, this architecture may be sufficient to achieve the goal of SSMT systems. However, due to the limitations of the current state-of-art technology in ASR and MT areas, there are errors or ambiguities in the output of these components, which are propagated to its successive components.

It is pretty evident from Fig. 1 that the primitive block is an ASR system. An ASR system's task is to convert speech in the source language to its corresponding text. The performance of the SSMT system majorly depends on the ASR system; if the output of the ASR system is erroneous, then it is likely to propagate throughout the pipeline, which results in an incorrect speech-to-speech translation. Therefore, the speech recognition [8] component must be concerned less with transcription fidelity than semantic fidelity, while the MT component must try to capture the meaning or intent of the input sentence [9–15] without being guaranteed a syntactically legal sequence of words. In this chapter, we have broadly discussed about two of the projects which we have worked on, which are namely,

1. "Speech to Speech Translation & performance measurement platform for Broadcast Speeches and Talks funded by TDIL MeitY, Government of India. (**June 2019 to March 2020**) **Status: Completed**"

2. "Crowd Sourcing of Large Speech Data Sets To Enable Indian Language Speech - Speech Solutions (Pilot Project) funded by TDIL MeitY, Government of India.
(**October 2020 to October 2021**) **Status: In progress (As on date September 29, 2021)** "

So as a part of this chapter, initially, we have drawn our experience in the functioning of the SSMT pipeline and later strategies involved in collecting a crowdsourced speech corpus for the speech recognition task have been discussed.

The remaining paper is organized as follows: Section 2 briefly describes the Speech to Speech Translation & Performance Measurement Platform for Broadcast Speeches & Talks and analysis on the pipeline. In Sect. 3, the authors describe the CSTD-Telugu Corpus: Crowd-Sourced Approach for Large-Scale Speech data collection and its experimental results. The authors conclude the paper and give possible future directions in Sect. 4.

2 Speech to Speech Translation and Performance Measurement Platform for Broadcast Speeches and Talks

In this work, we demonstrate the capability of technologies for performing speech to speech translation in Indian context in an industry grade technology stack and capture performance benchmarks to enable industry application grade technology development. Speech to speech translation is performed using speech recognition, machine translation and text to speech. In multilingual society like India, speech to speech translation has plenty of potential applications like educational videos translation, overcoming the language barrier for communication and learning etc. In Indian context it is challenging because of lack of proper speech recognition, machine translation and text to speech systems for Indian languages. In this project we will convert Prime Minister Maan Ki Baat Hindi speech to English and Telugu automatically. The same will be extended to other speech to speech application needs expressed by broadcast industry e.g. sports commentary, multi-lingual interviews etc.

Observation and Analysis. Speech-to-speech translation systems have been developed for converting one language speech into another language speech, with the goal of helping people who speak different languages to communicate with each other. As mentioned earlier, such systems have usually been broken into three separate components they are as follows:

1. Automatic Speech Recognition (ASR)
   – to transcribe the source speech to text The performance evaluation metric for an ASR system is calculated in terms of Word Error Rate (WER).

$$WER(\%) = \frac{S + I + D}{N} \times 100$$

where Insertion - Ins (I), Deletion - Del (D), Substitution - Sub (S)

2. Machine Translation (MT)
   - to translate the transcribed text into the target language. In this project we have considered human evaluation by judges on a scale of 4 point scale.[1]
     1. **Unacceptable:** Certainly incomprehensible and/or little or no information is accurately transferred.
     2. **Possibly Acceptable:** Perhaps comprehensible (given enough context and/or time to work); Some information has certainly been transferred.
     3. **Acceptable:** Not perfect (stylistically or grammatically odd), but certainly perceptible and with perfect transfer of all important information.
     4. **Ideal:** Not an accurate translation, but grammatically correct and accurately transmitted with all information.
3. Speech synthesis (Text-to-Speech)
   - to generate speech in the target language from the translated text. In this project we have considered human evaluation by judges on a scale of 5 point scale. (See footnote 1)
     1. **Bad:** Totally unacceptable, unintelligible, annoying.
     2. **Poor:** Sounds very unnatural, weird, many issues like noise, shaky, muffling but intelligible.
     3. **Fair:** Robotic, some minor issues like noise, shaky, muffling, but overall is acceptable.
     4. **Good:** Natural and close to human beings.
     5. **Excellent:** Very natural and sounds like a human being.

Dividing the task into such a cascade of systems has been very successful, powering many commercial speech-to-speech translation products. Following are the set of observations which we have observed while integrating:

- **Steps followed in Human Evaluation**

S1. The Audio is passed through the respective ASR system. As discussed prior the main job of an ASR is to convert the given audio into corresponding text. So once text is obtained we calculate the Word Error Rate (WER) from the ground truth (reference) transcripts.

* Following are statistics of sessions which we have considered for ASR, MT and synthesis analysis for respective APIs (Google, MSR, CDAC, and IITM):

Session June 2019	
Audio duration (HH:MM:SS)	00:30:59
Number of sentences	213
Number of words	4516

[1] This metric has been provided by Microsoft - IDC Hyderabad.

Session July 2019	
Audio duration (HH:MM:SS)	00:25:34
Number of sentences	390
Number of words	3588
Session August 2019	
Audio duration (HH:MM:SS)	00:31:34
Number of sentences	425
Number of words	4348

Consider the following examples:
Example1

Actual utterance / Reference

'मन क बात' हमेशा क तरह, मेर तरफ से भी और आपक तरफ से भी एक प्रतीक्षा रहती है ।

ASR Output

मन क बात हमेशा क तरह, मेर तरफ से भी और आपक तरफ से भी एक DEL रहती है ।

DEL - Deletion

So, in the above utterance the number of deletion is 1

So now we have to make a note of the number of deletions, number of substitutions and number of insertions from the ASR outputs.

S2. Later the text from the ASR output is fed to the translation system and the translation output is given to human's for evaluation. And the scores have been given by them. So the number of participants in this exercise are 15 and the average of all these 15 participants has been reported. The performance evaluation is done to the scale of 4.

S3. The translated text is taken and fed to the respective speech synthesis engines for the generation of text to speech. So once we had the synthesized output we performed subjective evaluation with 15 participants and asked to grade the outputs to the scale of 5 and mean of which is reported here.

The platform which is developed has been integrated using multiple API's and it is observed the stability of it mainly depends on their server (Parent node) (Tables 1 and 2).

Actual utterance / Reference

हमेशा की तरह, मेरी तरफ से भी और आपकी तरफ से भी एक प्रतीक्षा रहती है। इस बार भी मैंने देखा कि बहुत सारे पत्र, comments, phone मिले हैं –ढेर सारी कहानियां हैं, सुझाव हैं, प्रेरणा हैं

MSR ASR Output

हमेशा की तरह, मेरी तरफ से भी , और आपकी तरफ से भी, एक प्रतीक्षा रहती है | इस बार भी, मैंने देखा कि भाव सारे पत्र, कॉमेंट्स, फोन मिले DEL , ढेर सारी कहानियां हैं | सुझाव है | प्रेररणा है |

Google ASR Output

हमेशा की तरह मेरी तरफ से भी और आपकी तरफ से भी एक प्रतीक्षा रहती है इस बार भी मैंने देखा कि DEL सारे पत्र कॉमेंट्स फोन DEL मिले DEL ढेर सारी कहानियां हैं सुझाव है प्रेरणा है

CDAC ASR Output

हमेशा की तरह मेरी तरफ से भी और आपकी तरफ से भी एक प्रतीक्षा रहती है इस बार भी मन देखा क DEL बहुत सारे प DEL कम स पखं SUB DEL मले DEL ढेर सार DEL कहा नयाँह सझुाव है ◌DELरणा है

Table 1. Summary of all the API's used and metrics used in calculating the performance evaluation is as follows:

S. No	Sessions	API's	Metrics			
			WER (%)	MT (/4)		TTS (/5)
				Hi-Te	Hi-En	
1.	June 2019	Google	9.4	3.3	3.1	3.1
		MSR	6.0	3.4	3.41	
		CDAC	9.2	–	–	–
		IITM	–	–	–	2.7
2.	July 2019	Google	9.54	3.2	3.1	3.1
		MSR	6.2	3.4	3.4	3.41
		CDAC	9.52	–	–	–
		IITM	–	–	–	2.7
3.	August 2019	Google	9.2	3.3	3.2	3.1
		MSR	6.1	3.41	3.4	3.41
		CDAC	9.52	–	–	–
		IITM	–	–	–	2.7

2.1 Challenges Faced While Building the SSMT Pipeline

The two verticals, namely, speech processing and text processing, need to be carefully handled while performing any task related to transcription or translation. Among both, text processing has wider exposure and is better understood.

Table 2. Summary of all the APIs used and latency involved in each component and the number which is reported below have been tested on 100 Mbps internet speed.

S. No	Sessions	API's	Latency (mm:ss)			Total
			ASR	MT	TTS	
1.	June 2019	Google	00:06	00:08	00:14	00:28
		MSR	00:08	00:08	00:20	00:36
		CDAC	00:06	–	–	–
		IITM	–	–	00:26	–
2.	July 2019	Google	00:06	00:08	00:14	00:28
		MSR	00:08	00:08	00:20	00:36
		CDAC	00:06	–	–	–
		IITM	–	–	00:26	–
3.	August 2019	Google	00:06	00:08	00:14	00:28
		MSR	00:08	00:08	00:20	00:36
		CDAC	00:06	–	–	–
		IITM	–	–	00:26	–

Pointing this out, text translation itself is an arduous task wherein it has to handle the nuances of the target language. In literature, few of the groups have tried to automatically translate spoken language into its target language text and reported that apart from loss in the context information, it also involved difficulty while handling semantics, domain context, disfluency (repairs, prolongations, false starts, etc.), dialog effect and few more uncertainty.

Issues: As discussed above, the difficulties involved in both speech recognition and text translation, people have made a few attempts to solve and bring naturalness into the pipeline. Few of them are,

Speaking Style (Read Speech vs Conversational). In practice, the articulation of speech, place a significant role in building speech-based products. In read speech mode, it is found that there will not be hesitations or disfluencies as most of it is prompted carefully. Whereas, in the case of conversational speech, such kind of behavior will not be observed. Given such information to the system, while building, it produces high accuracy. For example, News channels are considered as read speech and meeting scenarios comes under conversational speech.

Pacing (Consecutive vs. Simultaneous). Pauses are very common in the natural mode of communication, which help segregate the utterances spoken while producing the translations by the systems. This type of mechanism comes under consecutive mode. Therefore it eases the process of translation. In the SSMT pipeline, recognition and translation engines are expected to perform in synchronous mode while the speaker speaks. This type of mode is said to be a simultaneous mode.

Speed and Latency (Real-Time vs. Delayed Systems). Problems may persist in the SSMT pipeline depending on the speed requirements and the waiting time, which in general is called latency. In the synchronous mode, an optimal threshold value should be considered so that the speaker is in line with the output of the system. Consider the scenario where the lecture is being delivered or live streaming program, where the SSMT output is indeed expected to have low latency. But in standalone cases(post-hoc, viewing, or browsing), there is no need to worry about the speed and latency of the system.

Microphone Handling. In general, speakers tend to use microphones close to the mouth, which yields a clear speech signal like mobile phones. It is noticed that performance degradation is observed when speakers are far away from the microphone as it captures external or background noise.

Human Factors and Interfaces: Speech translation facilitates human communication in all ways, so a human interface is required. In a perfect world, we need to hear and comprehend that the discussion accomplices communicate in our language and don't have an interpretation program: the errand of the interface is to make the language hindrance as straightforward as could be expected. We need greatest speed and least obstruction from one viewpoint while keeping up with most extreme precision and effortlessness on the other. These are competitive goals. Better interface solutions help balance them; But no definitive solutions are expected in the near future, as even human commentators generally spend considerable time on clarity dialogues. As long as the exact accuracy is unclear, effective error recovery mechanisms are desirable. The first step is to enable users to identify errors in speech recognition and translation. Once the errors are found, mechanisms are needed for correction and then for adaptation and improvement. Literate users can detect errors generated by the speech recognition engine on the device screen. Text-to-speech playback of ASR results is used (but rarely used so far) for illiterate users or to initiate blind use. Some systems may allow users to type the wrong word or write it by hand to correct ASR errors. Facilities may be provided instead for voice-based correction (although these are also rarely used). The whole input may be repeated, but the same errors may be repeated, or new ones may be triggered. At last, multimodal goals can be upheld, for instance, in the realistic interface with manual determination of the mistake and voice correction. In any case, if a fragment or elocution can be revised, it very well may be shipped off machine interpretation (essentially in frameworks where ASR and MT parts are exact). Then, at that point, the distinguishing proof and adjustment of the interpretation results can be worked with. In ASR or MT, blunders are irritating, however repeating mistakes are extremely irritating. In a perfect world, frameworks ought to gain from their missteps with the goal that mistakes over the long haul and being used are limited. On the off chance that AI or some neural model is made accessible, clients should exploit any remedies provided. Then again, intelligent update components might be given.

Neural Speech to Speech: SSMT is to decipher discourse from one language into another. The neural model is useful for separating correspondence hindrances between individuals who don't share a typical language. In particular, it is feasible to prepare models to achieve the undertaking straightforwardly without depending on transitional message portrayal. It is rather than customary SSMT frameworks, which can be comprehensively grouped into three sections: Automatic Speech Recognition (ASR), Text-to-Text Machine Translation (MT), and Text-Speech (TTS) combination. Course frameworks have the likely issue of blunders between parts, e.g., distinguishing proof mistakes prompting more huge interpretation blunders. SSMT models keep away from this issue through preparing to address task-to-end. They additionally enjoy upper hands over course frameworks as far as diminished computational necessities and lower inductance delay, as just one disentangling step is needed rather than three. In addition, direct models can naturally retain translingual and non-linguistic information during translation. Finally, direct conditioning of the input speech makes it easier to learn to create clear pronunciation of words that do not require translation, such as names. End-to-end training can be given to the direct speech-to-speech translation format. Multi-task training, especially with speech-to-text tasks, facilitate training without pre-defined settings to influence high-level representations of source or target information in the form of transcripts. However, intermediate text representation is not used during inference. End-to-end architecture can be developed on a sequence network based on a vocoder that converts attention-based sequences (the ability to translate speech into speech) and target spectrograms into time-domain waveforms.

In this work, we have integrated API's from different vendors and developed a platform for evaluating the performance measure on broadcast speeches. As a part of it we have found that though Microsoft API's have a bit large latency when compared to the CDAC and Google it is producing the accurate outcomes. So among the lot Microsoft stands first next CDAC-Pune (for ASR) and later Google (Not adopted for MKB data). Later few challenges related to pipeline has been discussed. The demo of this project can be found here[2].

3 CSTD-Telugu Corpus: Crowd-Sourced Approach for Large-Scale Speech Data Collection

The availability of speech databases for Indian languages is minimal. Most of the Indian languages are spoken widely throughout the global still. There exist no annotated speech databases for building reliable speech technology products. To bridge this gap, people in the literature have adopted crowdsourcing for collecting such data collaboratively. In this project, we describe an experience of Telugu speech database collection for building automatic speech recognition tasks. It was done in collaboration with Ozontel Technologies private limited, Pacteraedge, and CIE IIIT-Hyderabad. In other sections, we explain the platform we have developed and strategies adopted for collecting the corpus.

[2] https://drive.google.com/file/d/1Xu0ELaHtgXRXulwf-FqV_6qZHv6BCghR/view.

3.1 Overview of the Pipeline

The framework has been developed so that any person can log into it by providing the basic information and start contributing his/her speech to the platform. In this process, people can either use their laptops or mobile phones as a medium for recording. The main advantage of crowdsourcing is that anybody can contribute from anywhere, sitting across the globe. So in this project, our collaborators have reached and pooled an audience for the data collection. The setup used for this crowd-sourced data collection is shown in Fig. 3. An upper bracket of 90 min is allotted for a speaker to provide his/her speech. Before starting the recording, each of them is provided with the guidelines to be followed like, and they should speak clearly, distinctly, naturally with few filler words. Once the

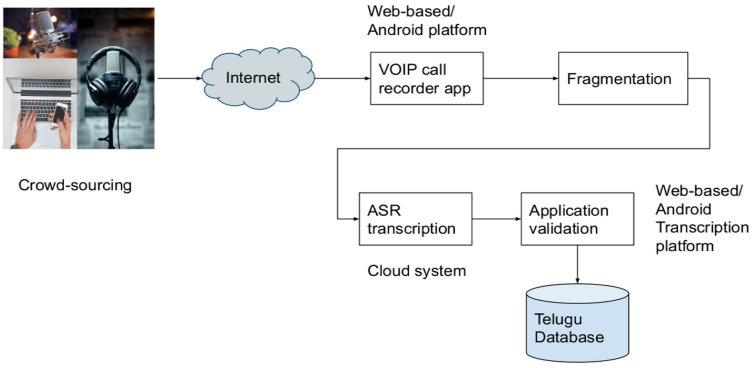

Fig. 2. Crowd-sourced set-up for Telugu speech data collection.

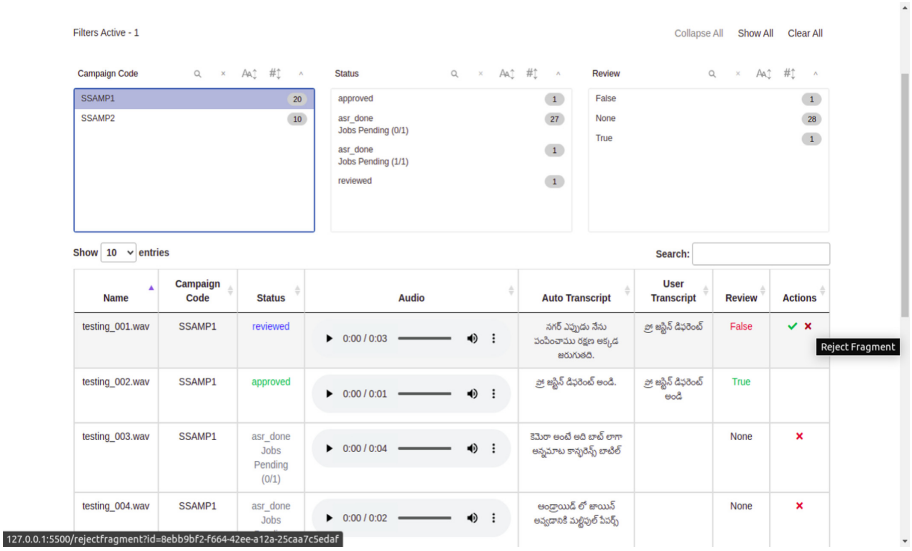

Fig. 3. Admin dashboard of the platform.

speech is captured through the VOIP interface, the platform make sure that the recorded speech is 16 Khz of sampling rate and 16 bit (Fig. 2).

The platform is built in such a way where users can select his/her topic of interest, which would be displayed on the user's screen. It also can handle multiple users simultaneously. Once the recording is done, the speech data is sent to the back-end server, and necessary formatting is done before sending it for further processing. Once the data is formatted, it is sent for fragmentation so that it could be used for building an ASR system building. The fragmentation algorithm is written in such way that it identifies non-speech region and chops the audio files with respective it. It also makes sure that the selected fragment is within the range of 3–15 seconds and fragments which doesn't comes under this category (specified duration) are discarded. The accepted fragments are passed through the ASR to generate rough transcripts so that it is passed human validators to verify the transcripts. These transcripts undergo a two level verification process by human transcribers so that transcripts will be error free.

In this task we have focused on creating a crowdsourcing platform for handling large-scale speech data collection of Telugu language. As a part of this task we have collected a good quality of crowd sourced Telugu speech database. Experience of the entire project will be discussed in out future paper.

4 Conclusion and Future Work

In this work, we have demonstrated the capability of technologies for performing speech to speech machine translation in Indian context. Later challenges faced while building the SSMT pipeline are broadly discussed. The platform which we have developed to collect a crowdsource speech corpus for speech recognition task is briefly explained.

Further progress awaits the maturity of vital components of any speech translation system - speech recognition, machine translation, speech synthesis, and practical infrastructure. In cases of demand, facilities are required to quickly switch between or interleave between automatic and human interpreters and tools to assist those interpreters. We also need feedback tools to assure customers that more human (and expensive) intervention is helpful. Serious R&D for speech-to-speech machine translation has to be continued worldwide, both with and without government sponsorship.

Acknowledgements. We would like to acknowledge the Technology Development for Indian Languages (TDIL), Ministry of Electronics and Information Technology (MeitY), Government of India, for allowing us to work on the pilot project for collecting large-scale Telugu corpus. We also thank Indian Institute of Technology-Madras Speech Lab, Microsoft IDC-Hyderabad, CDAC-Pune for providing there respective APIs for benchmarking the platform which we have developed. We would also like to thank our collaborators Ozonetel, Pacteraedge, and Swecha for Crowd Sourcing of Large Speech Data Sets To Enable Indian Language Speech - Speech Solutions (Pilot Project).

References

1. Anumanchipalli, G.K., Oliveira, L.C., Black, A.W.: Intent transfer in speech-to-speech machine translation. In: 2012 IEEE Spoken Language Technology Workshop (SLT), pp. 153–158. IEEE, December 2012
2. Zhang, R., Kikui, G., Yamamoto, H., Soong, F.K., Watanabe, T., Lo, W.K.: A unified approach in speech-to-speech translation: integrating features of speech recognition and machine translation. In COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, pp. 1168–1174 (2004)
3. Vilar, D., Xu, J., Luis Fernando, D.H., Ney, H.: Error analysis of statistical machine translation output. In: LREC, pp. 697–702, May 2006
4. Hashimoto, K., Yamagishi, J., Byrne, W., King, S., Tokuda, K.: Impacts of machine translation and speech synthesis on speech-to-speech translation. Speech Commun. **54**(7), 857–866 (2012)
5. Matusov, E., Kanthak, S., Ney, H.: On the integration of speech recognition and statistical machine translation. In: Ninth European Conference on Speech Communication and Technology (2005)
6. Frederking, R.E., Black, A.W., Brown, R.D., Moody, J., Steinbrecher, E.: Field testing the tongues speech-to-speech machine translation system. In: LREC, May 2002
7. Tomokiyo, L.M., Peterson, K., Black, A.W., Lenzo, K.A.: Intelligibility of machine translation output in speech synthesis. Presented at the Ninth International Conference on Spoken Language Processing (2006)
8. Salesky, E., Sperber, M., Black, A.W.: Exploring phoneme-level speech representations for end-to-end speech translation (2019). arXiv preprint arXiv:1906.01199
9. Carbonell, J.G., Lavie, A., Levin, L., Black, A.: Language technologies for humanitarian aid (2005)
10. Levin, L., et al.: The Janus-III translation system: speech-to-speech translation in multiple domains. Mach. Transl. **15**(1), 3–25 (2000)
11. Waibel, A., et al.: Speechalator: two-way speech-to-speech translation in your hand. In: Companion Volume of the Proceedings of HLT-NAACL 2003-Demonstrations, pp. 29–30 (2003)
12. Schultz, T., Alexander, D., Black, A.W., Peterson, K., Suebvisai, S., Waibel, A.: A Thai speech translation system for medical dialogs. In: Demonstration Papers at HLT-NAACL 2004, pp. 34–35 (2004)
13. Suebvisai, S., Charoenpornsawat, P., Black, A., Woszczyna, M., Schultz, T.: Thai automatic speech recognition. In: Proceedings (ICASSP'05) IEEE International Conference on Acoustics, Speech, and Signal Processing 2005, vol. 1, pp. I–857. IEEE, March 2005
14. Wilkinson, A., Zhao, T., Black, A.W.: Deriving phonetic transcriptions and discovering word segmentations for speech-to-speech translation in low-resource settings. In: INTERSPEECH, pp. 3086–3090 (2016)
15. Scharenborg, O., et al.: Building an ASR system for a low-resource language through the adaptation of a high-resource language ASR system: preliminary results. In: Proceedings of ICNLSSP, Casablanca, Morocco (2017)
16. Davis, S., Mermelstein, P.: Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoust. Speech Signal Process. **28**, 357–366 (1980)
17. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**, 257–286 (1989)

18. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning - Data Mining, Inference, and Prediction. SSS, Springer, New York (2009). https://doi.org/10.1007/978-0-387-84858-7
19. Freitas, J., Calado, A., Braga, D., Silva, P., Dias, M.: Crowdsourcing platform for large-scale speech data collection. In: Proceedings FALA (2010)
20. Jyothi, P., Hasegawa-Johnson, M.: Acquiring speech transcriptions using mismatched crowdsourcing. In: Proceedings of the AAAI Conference On Artificial Intelligence, vol. 29 (2015)
21. Butryna, A., et al.: Google crowdsourced speech corpora and related open-source resources for low-resource languages and dialects: an overview. ArXiv Preprint ArXiv:2010.06778 (2020)
22. Arora, K., Arora, S., Roy, M., Agrawal, S.: Multilingual crowdsourcing methodology for developing resources for under-resourced Indian languages
23. Chopra, M., Medhi Thies, I., Pal, J., Scott, C., Thies, W., Seshadri, V.: Exploring crowdsourced work in low-resource settings. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1–13 (2019)
24. Prasad, K., Virk, S., Nishioka, M., Kaushik, C.: Crowd-sourced technical texts can help revitalise Indian languages. In: Proceedings Of LREC 2018, Workshop WILDRE4, pp. 11–16 (2018)
25. Jonell, P., Oertel, C., Kontogiorgos, D., Beskow, J., Gustafson, J.: Crowdsourced multimodal corpora collection tool. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (2018)
26. Arora, S., Arora, K., Roy, M., Agrawal, S., Murthy, B.: Collaborative speech data acquisition for under resourced languages through crowdsourcing. Procedia Comput. Sci. **81**, 37–44 (2016)
27. Abraham, B., et al.: Crowdsourcing speech data for low-resource languages from low-income workers. In: Proceedings of the 12th Language Resources and Evaluation Conference, pp. 2819–2826 (2020)
28. Srivastava, B., et al.: Interspeech 2018 low resource automatic speech recognition challenge for Indian languages. In: SLTU, pp. 11–14 (2018)
29. Bell, L., Boye, J., Gustafson, J.: Real-time handling of fragmented utterances. In: Proceedings NAACL Workshop on Adaptation in Dialogue Systems, pp. 2–8 (2001)
30. Yang, Z., Liu, W., Jiang, W., Hu, P., Chen, M.: Speech fragment decoding techniques using silent pause detection. In: Chinese Conference on Pattern Recognition, pp. 579–588 (2012)
31. Povey, D., et al.: The Kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (2011)
32. Gales, M.: Maximum likelihood linear transformations for HMM-based speech recognition. Comput. Speech Lang. **12**, 75–98 (1998)