Developing a Framework for Acquisition and Analysis of Speeches

Md. Billal Hossain(✉), Mohammad Shamsul Arefin,
and Mohammad Ashfak Habib

Department of Computer Science & Engineering, Chittagong University of
Engineering & Technology, Chattogram 4349, Bangladesh
mhbillal160@gmail.com
{sarefin,ashfak}@cuet.ac.bd

Abstract. Speech plays a vital role for human communication. Proper delivery of speech can enable a person to connect with a large number of people. Nowadays, a lot of valuable speeches are being provided by many popular people throughout the world and it will be very helpful if important information can be extracted from those speeches by analyzing them. An automatic speech-to-text converter can facilitate the task of speech analysis. There have been carried out a lot of works for the conversion of speech to text in the last few decades. This paper presents a framework for the acquisition of speech along with the location of the speaker and then conversion of that speech into text. We have worked with speeches containing three different languages. To evaluate our framework, we collected speeches from several locations and the result shows that the framework can be used for efficient collection and analysis of the speeches.

Keywords: Recording · Location tracking · Database · Speech recognition

1 Introduction

Speech is the most natural form of communication and interaction between humans. It enables us to exchange knowledge without experiencing it directly. It is a very important part of human development. Speech can convince its audience to some particular agenda [1]. To attract and motivate people speeches are delivered by many people. Good, accurate and authenticate speeches can guide people to the right direction and enrich their knowledge. It can play an effective role in changing the mentality of large number of people or strengthening their belief in the speaker. For example, we can think about the speech given by Abraham Linclon [2] or Martin Luther King [3] in order to inspire their people. They used their motivational speaking skill for spreading their vision.

C. R. Panigrahi et al. (eds.), *Progress in Advanced Computing and Intelligent Engineering*, Advances in Intelligent
Systems and Computing 1198, https://doi.org/10.1007/978-981-15-6584-7_6

However, misguided speeches can lead to a crime and is a threat to the country. It often directs people in the wrong direction and increases risks in the society. Due to the rapid growth of internet technology, any misleading speech can be easily distributed among different groups of people and by the influence of that speech they can create huge problems for the society. It does not only threaten the life and livelihood of the people of a country but also interrupt in the peacekeeping throughout the world. This types of threats have got the concern in both national and international security and steps are being taken to stop such types of crimes.

So, a good speech repository and speech monitoring system can assist in spreading valuable speeches and prevent spreading any misleading information. By observing the speeches provided by the speakers, it is possible to stop any potential crime. Besides that, if the audio of speech can be converted into text then the speech can be processed automatically to extract information from that text which will also help many search engines that uses computer readable text as data and offers large amount of data ranked by similarity.

However, due to difficulties in recording and varieties of speeches, it is difficult to accurately extract information from these speeches. In this paper, a framework is proposed for the acquisition of speech efficiently. The location of the speaker is also tracked. Then this speech is stored in a database as an audio file along with the location information. After that, the language of the speech is detected and then the speech is converted into text using a speech recognition API. We considered three different languages in our framework: English, Bengali and Arabic.

The remaining section of the paper is organized as follows, Sect. 2 shows the previous work related to this paper, Sect. 3 describes about the methodology of the proposed framework, Sect. 4 represents the implementation and performance evaluation of the system and in Sect. 5, a conclusion is drawn with future scope of improvements.

2 Related Work

The usage of mobile phone has increased drastically over the last few years. It is approximately 3.5 times larger than PCs [4]. Nowadays, mobile phone is not only being used as a tool for making call and writing SMS, but also act as a mean for personal entertainment and communication with the world [5]. Almost every features that is available in PC, can also be found in a smartphone. Smartphones are available to almost everyone and one of the most popular operating systems being used in those devices is Android, developed by Google [6]. Even a normal android device can perform varieties of tasks [7], like recording audio or video, capturing image, detecting location etc. Android applications can be developed by using these features which can be applied in many fields [8]. For example, location service and audio recording feature of the android smartphone can be used for efficient recording of speech along with the location information for effective analysis of the speech that is provided by many valuable speakers around the world.

Analysis of speech manually is a lengthy process and requires a lot of time. Speech recognition or simply known as speech-to-text conversion may facilitate this task, since analysis of text is lot easier than the direct analysis of speech. Recently, many works have been carried out in this field and got a lot of improvements [9,10]. Nowadays, Automatic speech recognition systems are quite well established and can operate at accuracy of more than 90% [11]. The advancement of algorithms and modern processors enabled the optimization of almost all the phases involved in the speech recognition task [12]. There are many commercial and open source systems like CMU Sphnix, Microsoft Speech API, AT&T Watson, Google Speech API etc. [13].

CMU Sphnix [14] is an open-source speech recognition system that was developed at Carnegie Mellon University (CMU). It consists of large vocabulary and speaker independent speech recognition codebase which is available to download and use. It has different versions and packages for different applications. The latest version uses Hidden Markov Model (HMM) [15] and a strong acoustic model that is trained with a large set of vocabulary. In [16], Pytorch-kaldi toolkit is used for automatic speech recognition. Microsoft has developed an speech recognition API known as Speech Application Programming Interface or SAPI for Windows applications [17]. It used Context Dependent Deep Neural Network Hidden Markov Model (CD-DNN-HMM) that is trained with a huge volume of dataset for achieving a good recognition accuracy. Currently it is being used in Microsoft Office, Microsoft Speech Server and Microsoft Agent.

Google has developed their own speech-to-text (gSTT) conversion API [18], using deep learning techniques that achieved less than 8% word error rate. This API is being used in many applications like voice search, voice typing, translation, navigation, YouTube transcription etc. Among all the above described systems, gSTT API gives better result than others in terms of both word error rate [19] and conversion time [20]. The conversion of speech-to-text in Google speech API takes place as soon as it receives the first voice packets which saves the time for further processing. For these reasons, Google speech API is chosen in our framework for the automatic conversion of speech into text.

3 Methodology

The framework is divided into three modules: (i) Speech Acquisition Module, (ii) Speech Storage Module, and (iii) Speech Recognition Module. In Fig. 1 the overall graphical structure of the framework is shown. First the speech will be recorded along with the location information. Then this speech will be stored in a database where it will be processed for automatic speech recognition. There is also a user friendly website where anyone can see all of the recorded speeches along with their location information. Since the number of speeches will be increased with the progression of time, there has been provided a way to filter the total number of speeches by searching among the speeches in the database based on various criteria.

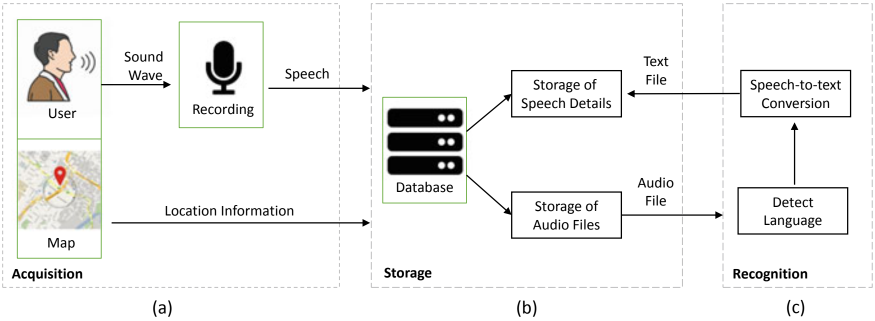


Fig. 1. Overall graphical structure of the system: **a** Speech acquisition module; **b** Speech storage module; **c** Speech recognition module

3.1 Speech Acquisition Module

Speech acquisition contains two parts. One for recording of speech and another for tracking the location. Figure 2 shows the architecture of the speech acquisition module. A Microphone is used to capture the sound of the speaker and it is stored as an audio file in the local storage. To track the speaker's location first latitude and longitude need to be calculated. Using the latitude and longitude we can calculate the actual address of the speaker. When the speaker finishes his speech, it is uploaded into the cloud database where further processing takes place.

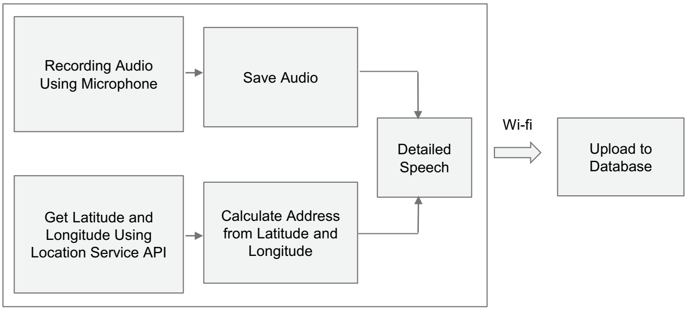

Recording Audio Using Microphone
Save Audio
Detailed Speech
Get Latitude and Longitude Using Location Service API
Calculate Address from Latitude and Longitude
Wi-fi
Upload to Database

Fig. 2. Framework architecture of the speech acquisition module

3.2 Speech Storage Module

Table 1 shows the structure of the table in the database. For each incoming audio file a row is created in the database with a unique identifier. The file name of the incoming audio file may be same as some other audio files in the database.

So, the file name is renamed as: Unique ID + '.' + File Extension. Initially, we do not know what is the language and the text contained in that audio. So, they are set to null. After analyzing the audio file we update the database table with its corresponding language and text file. The name of the text file is same as the audio file name except that its extension is set as 'txt'.

Table 1. Structure of the table in the database

unique_id	audio_file_name	Location	Language	text_file_name
001	001.wav	Pahartoli, Chattogram, 4349, Bangladesh	Bengali	001.txt
002	002.wav	Habiganj, Sylhet, 3310, Bangladesh	English	002.txt
003	003.wav	Durgapur, Chandpur, 3640, Bangladesh	Arabic	003.txt
004	004.wav	Dhanmondi, Dhaka, 1208, Bangladesh	Mixed	004.txt

From Table 1 it can be seen that, the audio_file_name and text_file_name can be obtained from the unique_id. So we can omit this extra two columns. Besides that, instead of using full name of the languages, we can encode them with numeric digits. i.e: 1 for Bengali, 2 for English, 3 for Arabic and 0 for mixed language (that contains one or more language in the speech). The updated database table is shown in Table 2.

Table 2. Structure of the updated table in the database

unique_id	Location	Language
001	Pahartoli, Chattogram, 4349, Bangladesh	1
002	Habiganj, Sylhet, 3310, Bangladesh	2
003	Durgapur, Chandpur, 3640, Bangladesh	3
004	Dhanmondi, Dhaka, 1208, Bangladesh	0

3.3 Speech Recognition Module

In the recognition module the language of the incoming speech is detected and the speech is converted into text. Figure 3 shows the overall architecture of the conversion process. We use a speech recognition API for converting audio into text. But the problem is that for long duration of speeches the accuracy of the conversion is very low. To solve this problem we can either split the speech into smaller chunks of constant size or we can split them based on the silence presented in the audio. If we split by keeping a constant duration then a word within the speech might get splitted and that word will not be detected. So, we choose to split the speech based on the silence presented. It is possible because when we speak, we pause for a small duration after finishing a sentence. That means, we can consider each chunk as a sentence. The speech is splitted if the amplitude level in the audio is less than −16 dBFS for more than 0.5 s.

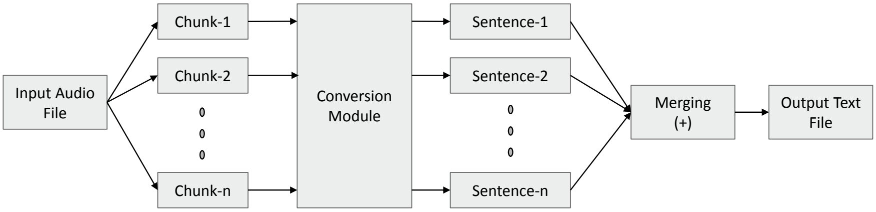


Fig. 3. Framework architecture of the speech recognition module

Another problem is that, we need to specify the language to which we want to convert it. For example, if a speech contains English language then we need to specify language parameter as English in the API. So, if we want to recognize speech containing multiple language we need to specify the language parameter manually. Besides that, most of the time speech contains more than one language and the system will not give good result for such type of mixed language speeches. To solve this, we send three requests in the API for three different languages for each sentence or chunk. Then the converted text is splitted into words and then checked in the dictionary of that language if that word exists. If the word is found in the dictionary then we increase the counter for that language. The language for which the counter value is maximum, is chosen as the language of that sentence. Algorithm 1 shows the process of detecting language and conversion of speech chunk into text. After the conversion the database is updated with the text file.

4 Implementation and Evaluation of the Framework

For implementing the framework we have developed an android application that can record speech along with the location information. We used Google speech-to-text (gSTT) converter as our speech recognition API. Three different languages: Bengali, English and Arabic are considered in our framework. For recording using android app an instance of the media recorder class is created and then the mic is initialized for the recording purpose. There are three buttons, first one for starting the recording, second one for stopping the recording and third one for uploading the recorded speech. The recording is started when

```
Input: $chunk$
Output: $language$, $text$
$max \leftarrow 0$;
for $i \leftarrow 1$ to 3 do
    $counter \leftarrow 0$;
    $converted\_text \leftarrow recognitionAPI(audio \leftarrow chunk, language \leftarrow i)$;
    for each word in $converted\_text$ do
        if word is in dictionary of language $i$ then
            $counter \leftarrow counter + 1$;
        end
    end
    if $counter > max$ then
        $max \leftarrow counter$;
        $language \leftarrow i$;
        $text \leftarrow converted\_text$
    end
end
```

Algorithm 1: Detecting language and converting speech chunk into text

the user presses the start recording button and then the recording of the speech is started and continues until the stop button is pressed. After the end of the recording the audio file is stored in the local storage of the phone. Now this speech can be uploaded by pressing the upload button. When the upload button is pressed the location of the device is tracked by using the location service of the android device and then it is uploaded to the server. To get the location first latitude and longitude need to be calculated which can be obtained by using Google Play Service. From the latitude and longitude, the actual address of the speaker is calculated which contains city, postal code, state and country name. When the uploading is complete, it will be available in the website and anyone can have access to that speech. After it is uploaded to the website it is passed to gSTT API for speech recognition.

To verify the proposed framework, we collected 100 speeches from 10 different locations in Bangladesh. The speeches were delivered by 10 people (10 speeches each) among which 6 of them were male and 4 of them were female. Each speaker installed the android app and gave permission to record their speech. After the speech is uploaded to the server it is verified with the actual data. Table 3 shows the performance of the speech acquisition process for 10 different locations. From the table it can be seen that some location information was not available in some region (represented by null). It is because of the variation in geocoding detail where the address line can vary. However, we can say that the acquisition of speech is done accurately with a percentage of 100%.

The performance of the speech recognition is dependent on the quality of the speech. If the environment where the speech is recorded is very noisy or if the sound of the speech is low then it gives comparatively low performance. Google API has automatic noise reduction in the recognition process which helped in

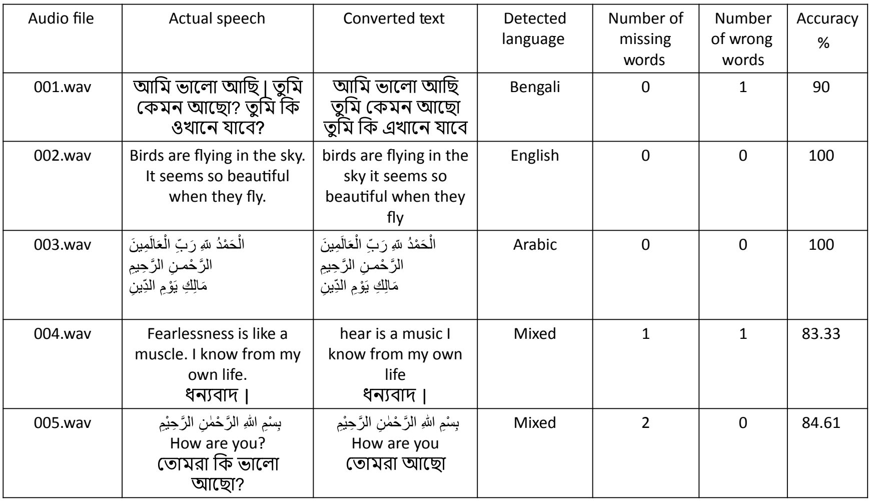
Table 3. Performance evaluation of speech acquisition process

Speaker	Actual location	Detected location	Speech quality
1	Pahartoli, Chattogram, 4349, Bangladesh	Pahartoli, Chattogram, 4349, Bangladesh	Highly satisfactory
2	Habiganj, Sylhet, 3310, Bangladesh	Habiganj, Sylhet, 3310, Bangladesh	Highly satisfactory
3	Durgapur, Chandpur, 3640, Bangladesh	Durgapur, Chandpur, null, Bangladesh	Satisfactory
4	Dhanmondi, Dhaka, 1208, Bangladesh	Dhanmondi, Dhaka, 1208, Bangladesh	Highly satisfactory
5	Gulshan, Dhaka, 1213, Bangladesh	Gulshan, Dhaka, 1213, Bangladesh	Highly satisfactory
6	Dinajpur, Rangpur, 5262, Bangladesh	Dinajpur, Rangpur, 5262, Bangladesh	Highly satisfactory
7	Jamalpur, Mymensingh, 2030, Bangladesh	Jamalpur, Mymensingh, 2030, Bangladesh	Highly satisfactory
8	Bogura, Rajshahi, 5892, Bangladesh	Bogura, Rajshahi, 5892, Bangladesh	Highly satisfactory
9	Barguna, Barisal, 8730, Bangladesh	null, Barisal, 8730, Bangladesh	Satisfactory
10	Bagerhat, Khulna, 9301, Bangladesh	Bagerhat, Khulna, null, Bangladesh	Highly Satisfactory

achieving a good recognition accuracy of the speech. Google speech recognition API achieved as low as 8% error rate in speech recognition [11]. Table 4 shows the performance of the recognition module. We can see that for speeches containing one language the recognition accuracy is comparatively higher and for mixed language the accuracy is also satisfactory if each sentence consists of a single language. However, if a sentence contains multiple language the system gives low recognition accuracy. For a speech if the total number of word is T and total number of missing word is M and total number of incorrect word is W, then the accuracy of the recognition is computed as,

$$\text{Accuracy} = \left(1 - \frac{M + W}{T}\right) \times 100\% \tag{1}$$

Table 4. Recognition accuracy of the framework for different languages



Audio file	Actual speech	Converted text	Detected language	Number of missing words	Number of wrong words	Accuracy %
001.wav	আমি ভালো আছি। তুমি কেমন আছো? তুমি কি ওখানে যাবে?	আমি ভালো আছি তুমি কেমন আছো তুমি কি এখানে যাবে	Bengali	0	1	90
002.wav	Birds are flying in the sky. It seems so beautiful when they fly.	birds are flying in the sky it seems so beautiful when they fly	English	0	0	100
003.wav	الْحَمْدُ للهِ رَبِّ الْعَالَمِينَ الرَّحْمـنِ الرَّحِيمِ مَالِكِ يَوْمِ الدِّينِ	الْحَمْدُ للهِ رَبِّ الْعَالَمِينَ الرَّحْمـنِ الرَّحِيمِ مَالِكِ يَوْمِ الدِّينِ	Arabic	0	0	100
004.wav	Fearlessness is like a muscle. I know from my own life. ধন্যবাদ ।	hear is a music I know from my own life ধন্যবাদ ।	Mixed	1	1	83.33
005.wav	بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ How are you? তোমরা কি ভালো আছো?	بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ How are you তোমরা আছো	Mixed	2	0	84.61

The overall feasibility of the proposed framework is shown in the Table 5. It can be seen that the framework is feasible for many applications and it may help in performing many tasks very efficiently.

Table 5. Overall feasibility of the proposed framework

Evaluation metric	Comments
Availability	Android phone is available to almost everyone
Implementation cost	Low
Recording quality	Very satisfactory
Location tracking	Nearly 100% accurate
Speech recognition	Less than 8% error rate
Applicability	Speech collection, speech monitoring, automation etc.

5 Conclusion

In this paper a framework is shown which can record speech and facilitate the task of speech analysis. Speech is recorded using an android app and the conversion of text is done by using Google speech recognition API. The system can be further applied in many fields like automatic subtitle creation, car driving using voice command, hearing impaired people etc. which can reduce a lot of manual works. Besides that, the collection of speech will enable people to access to the speech anytime from anywhere. It can also be used for suspicious content detection in the speech which may prevent from occurring many unexpected events in the society. For such importance of a speech recognition framework, it can be applied in many ways which might ease the way we lead our daily life.

Acknowledgments. This research is supported by the project **'Establishment of CUET IT Business Incubator'.**

References

1. Suedfeld P, Bluck S, Ballard EJ, Baker-Brown G (1990) Canadian federal elections: motive profiles and integrative complexity in political speeches and popular media. CJBS 22(1):26–36
2. Holzer H (2004) Lincoln at Cooper Union: the speech that made Abraham Lincoln president. Simon and Schuster
3. Vail M (2006) The "Integrative" Rhetoric of Martin Luther King Jr.'s' I Have a Dream" speech. Rhetoric & Public Affairs 9(1):51–78
4. Gandhewar N, Sheikh R (2010) Google android: an emerging software platform for mobile devices. IJCSE 1(1):12–17
5. Rice RE, Katz JE (2003) Comparing internet and mobile phone usage: digital divides of usage, adoption, and dropouts. Telecommun Policy 27(8–9):597–623
6. Kaur P, Sharma S (2014, March) Google android a mobile platform: a review. In: 2014 recent advances in engineering and computational sciences (RAECS), pp 1–5. IEEE, Chandigarh, India
7. Ableson F, Sen R, King C, Ortiz CE (2011) Android in action. Manning Publications Co
8. Rogers R, Lombardo J, Mednieks Z, Meike B (2009) Android application development: programming with the Google SDK. O'Reilly Media, Inc
9. Petridis S, Stafylakis T, Ma P, Cai F, Tzimiropoulos G, Pantic M (2018) End-to-end audiovisual speech recognition. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, Calgary, AB, Canada, pp 6548–6552
10. Krishna G, Tran C, Yu J, Tewfik AH (2019) Speech recognition with no speech or with noisy speech. In: ICASSP 2019–2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, Brighton, United Kingdom, pp 1090–1094
11. Isaacs D (2010) A comparison of the network speech recognition and distributed speech recognition systems and their effect on speech enabling mobile devices. Doctoral dissertation, University of Cape Town
12. Khilari P, Bhope VP (2015) A review on speech to text conversion methods. IJARCET 4(7)
13. Gaida C, Lange P, Petrick R, Proba P, Malatawy A, Suendermann-Oeft D (2014) Comparing open-source speech recognition toolkits. Technical Report of the Project OASIS
14. Sphnix API. https://cmusphinx.github.io/
15. Mukherjee S, Mandal SKD (2014) A Bengali HMM based speech synthesis system. arXiv preprint arXiv:1406.3915
16. Ravanelli M, Parcollet T, Bengio Y (2019) The Pytorch-Kaldi speech recognition toolkit. In: ICASSP 2019–2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, Brighton, United Kingdom, pp 6465–6469

17. Brown R (2008) Exploring new speech recognition and synthesis APIs in Windows Vista. Talking Windows, MSDN Magazine
18. Cloud Speech-to-text. https://cloud.google.com/speech-to-text/
19. Këpuska V, Bohouta G (2017) Comparing speech recognition systems (Microsoft API, Google API and CMU Sphinx). Int J Eng Res Appl 7(3):20–24
20. Assefi M, Liu G, Wittie MP, Izurieta C (2015) An experimental evaluation of apple Siri and google speech recognition. In: Proceedings of the 2015 ISCA SEDE, pp 1–6