# Automatic Pause Boundary and Pause Duration Detection for Text-to-Speech Synthesis Systems in Indian Languages

**Atish Shankar Ghone, Rachana Nerpagar, Pranaw Kumar, Bira Chandra Singh, Prakash B. Pimpale, Sasikumar M.**
Centre for Development of Advanced Computing, Mumbai, pin 400049, India
{atish, rachana, pranaw, bira, prakash, sasi}@cdac.in

## Abstract

Text-to-Speech synthesis systems are built using training over recorded speech and corresponding text. To achieve a natural sounding TTS system, it is very important to use correctly pause marked text data in the training phase. Similarly, pause marked text data along with duration of each pause is needed to build a pause prediction model. In this paper, we propose a method to mark the pauses automatically at appropriate positions in the text data corresponding to the recorded speech. This approach makes use of automatic speech recognition and text correction methods. With this approach, we save large amount of human effort without compromising much on the accuracy. We describe the experiments and results for three Indian languages: Hindi, Marathi, and Odia. The system can easily be extended to other languages.

## 1 Introduction

While developing text-to-speech synthesis (TTS) systems for a language, there is constant endeavor to achieve higher intelligibility and naturalness.

There are two widely used techniques to build a Text-to-Speech Synthesis system: 1) Unit selection based speech synthesis (Hunt and Black, 1996), and 2) Statistical parametric speech synthesis (Zen et al., 2009; Ghone et al., 2017). Time aligned prompt files are the primary input for development of both kinds of synthesis systems. Intelligibilityof TTS output primarily depends on the accuracy of these time aligned prompt files. These files are prepared using suitable segmentation algorithms like HMM (Prahallad et al., 2006), group delay (Prasad et al., 2004), hybrid segmentation (Shanmugam and Murthy, 2014), etc. and it requires speech (wave) and corresponding transcription text files as input. Hybrid segmentation works better for Indian Languages (Shanmugam and Murthy, 2014). Further, it is also observed that we get better alignment, if a grapheme notation (e.g., comma) corresponding to the silence region in speech file is inserted into the text file.

One of the important factors of naturalness of synthesized speech is the presence of pauses at appropriate places. Pause also affects the intended meaning of speech sound. Pause prediction model is used to generate TTS output with proper pauses. This model predicts the position of pauses in the text utterance, which has to be synthesized. There are two ways to build a pause prediction model: 1) Rule Based, and 2) Data Driven Statistical Method. The rule-based approach requires availability of linguistic tools like Part of Speech (POS) tagger, morphological analyzer, shallow parser, etc. However, there is a lack of these tools with reasonable accuracy for Indian languages. That is why the data driven approach is a better choice for most of the Indian languages (Ghosh and Rao, 2012
Sarkar and Rao, 2015). Pause marked text along with the duration information is used as training data to build the pause prediction model.

Apart from the above use cases, the pause marked text and the duration information would be primary resource for any research or study related to pause marking and pause duration analysis in NLP domain.

The text utterance of the speech data does not always reflect the actual pauses taken by the speaker, as while recording, the speaker takes many pauses even if there is no grapheme indication in the text file. In order to insert the grapheme notation to indicate a pause in text utterance, corresponding to the pause in recorded speech, it is necessary to carefully listen to the recorded speech and insert pause marks at appropriate places in the text manually. This is a time consuming and costly task, also prone to inconsistency. Apart from this, finding duration of the identified pause is also a tedious task.

Attempts have been made to insert pause marks in the text and find its duration by forced align-



*Proc. of ICON-2018, Patiala, India. December 2018* ©*2018 NLPAI*, pages 28–34

ment of the speech file with the text prompts using the HMM tool (Sarkar and Rao, 2015); (Nguyen. 2015). EHMM (Prahallad et al., 2006.) is a tool present in festvox to implement it. However, it is not so promising for Indian Languages, because it requires a great amount of post processing and manual correction. Similar observation has been made by Nguyen (2015): "EHMM could deal with pause insertion, but it often failed to predict the pause appearance or pause duration in the speech corpus".

In this paper, we describe our approach to identify pause boundary in the text using Automatic Speech Recognition (ASR) and text correction methods. The ASR system is built using the same speech and text data. We split the speech file into multiple segments using silence region as a delimiter. We pass each segment into ASR System to recognize the corresponding text. Once the text for each segment is recognized, we concatenate each segment using pause marker (comma) as the separator to regenerate the text sentence. As the output of our ASR system is not expected to be accurate, we need to do post processing of regenerated text sentence to correct it. Regenerated text sentence is corrected using a devised method and original text.

We also describe our analysis to decide the minimum silence duration to be considered as a pause. The same is used as the threshold to split the speech file.

We have integrated all the components together and released the bundle as an open-source tool under GPL license. This tool is available at: https://github.com/TTS-cdac-mumbai/.

The rest of the paper is organised as follows: Section 2 describes the ASR system and section 3 discusses silence duration analysis. In section 4, system flow has been presented and results & analysis is given in section 5. Section 6 concludes the paper with a brief summary of the work and future direction.

## 2 Pause Marking with ASR System

We develop the ASR system with text data and speech files that are created for TTS. We use kaldi toolkit (Povey et al., 2011) to build ASR system. It uses Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs) to model acoustic features. The language-building tool IRSTLM (Bertoldi and Cettolo, 2008) is used to build Language Model. We transformed the TTS data to the format required by kaldi toolkit. Unified parser (Baby et al., 2012) is used for lexicon preparation. It works for 9 Indian Languages, viz., Hindi, Marathi, Tamil, Telugu, Malayalam, Kannada, Guajarati, Odia, and Bengali. We split the speech files into multiple segments and pass each segment to the developed ASR system, which outputs the corresponding text. Further, the generated text segments are concatenated with pause mark (comma) inserted at the end of each segment. We observe that the output of ASR contains significant amount of errors primarily due to inaccuracy of ASR. Therefore, the ASR output is processed for correction.

For the correction process, we have two inputs: 1) Sentence generated using ASR output (we call this A) containing some errors, but with right pause mark information represented by a comma, and 2) Original sentence (we call this O) which was referred for speech recording, but without pause mark information. Using these, a correct sentence with right pause mark information is generated using the following method.

First, apply word level tokenization to the original sentence O. Using the tokenized output, word level n-grams are generated, where n starts from 1 and goes maximum up to the length of sentence. For example,

Sentence: *ABC PQR XYZ*

Generated n-grams: *ABC* (1-gram), *ABC PQR* (2-gram), *ABC PQR XYZ* (3-gram), *PQR* (1-gram), *PQR XYZ* (2-gram), *XYZ* (1-gram).

Then, the pause marked sentence A, generated using ASR output is split into independent chunks by considering the pause marker as a separator. Now each chunk is processed to find out the nearest n-gram from the list of n-grams using minimum character level edit distance as measured by Levenshtein (1966). Nearest n-grams found for all the chunk units are then combined using a pause marker (comma) to generate a correct sentence. This corrected sentence is expected to have correct text content and the pause information. The generated correct sentence is then compared with the original sentence by neglecting the pause markers. If the match indicates that the performed correction has generated the right sentence, then the process of correction ends. If the match fails, the process of correction is repeated using unmatched corrected sentence and the original sentence. In this second iteration of correction, the

only difference is in the process of finding a nearest n-gram from the list of n-grams for a chunk. This time instead of just the chunk under consideration, previous chunk is also pre-fixed to it as a context. This combined piece of string is now used to find the nearest n-gram for the purpose of correction. Upon finding the nearest n-gram, the pre-fixed chunk is removed from the same. This is expected to give us a correct n-gram, which we might have missed in the previous iteration. The following is pseudo code for the described method

***Input***

*orSentence =Original text sentence without all-pause mark information*
*asrOutPut = Erroneous text sentence generated using ASR output with right pause mark*

***Algorithm***

```
ngrams = createNgrams(orSentence)
    pPhrases = splitIntoChunks(asrOutPut)
    correctedSentence = blank
    FOR phrase IN pPhrases do
        correctedSentence = correctedSentence
                    + pauseMarker
                    + nearestNgramFromNgramList(phrase)
        IF correctedSentence without pauseMarkers doesn't
match orSentence
            asrOutPut =  correctedSentence
            pPhrases = splitIntoChunks(asrOutPut)
            correctedSentence = blank
            FOR phrase IN pPhrases do
                correctedSentence = correctedSentence
                        + pauseMarker
                        + (nearestNgramFromNgram-
List(previousPhrase+phrase)
                        - previousPhrase)
        RETURN correctedSentence
```

ASR system was trained with various duration of speech data and it is observed that it works properly even with just one hour of speech data.

Example:

- without comma marked sentence of Marathi language

मला वाटलं कां पाणी माइया हाता खालां श्वास घेत होते.

- Comma inserted using system

मला वाटलं**, ए** पाणी माइया हाता खालां, श्वास घेत होते.

- Final corrected sentence comma marked

मला वाटलं, **कां** पाणी माइया हाता खालां, श्वास घेत होते.

This algorithm works properly even there is repetition of word or phrase in sentences.

## 3 Silence Duration Analysis

Deciding the duration of silence, which can be considered as threshold (i.e., any silence shorter than it should not be considered as pause) is a non-trivial task. There are two ways to handle it: 1) Considering the threshold taken by other researchers, and 2) Trying to devise a method to decide threshold.

Vadapalli, et al. (2012) did an experiment by considering three different values, i.e., 25 ms, 50 ms, and 80 ms as threshold. Campione and Véronis (2002) tried to distinguish silent pause with occlusives and suggests to consider 200ms of silence duration as pause. It states: "Silent pause shorter than 200 ms are very difficult to discriminate from occlusives and taking them into account requires enormous manual effort." In order to verify the claim, we did the following analysis.

Hindi text content of five paragraphs detailed as in table 2 was prepared. Nine different speakers were asked to read aloud these contents in natural way and the same were recorded.

The recorded speech and corresponding text were given to three new persons. They were instructed to listen to the speech files and manually put pause mark on the text files wherever they feel appropriate. It is observed that minimum silence duration which was considered as pause by all three speakers is around 180-200 millisecond.

Distinction between pause and occlusive is also crucial. Occlusives are those silences which occur even within words as a beginning part of the stop consonants, e.g. in the word "vaakya", occlusive silence occurs in the beginning part of the phone [k]. We did analysis of our recorded data and found that occlusives of various lengths like, 60ms, 90ms, 136ms, 170ms are there. Some words having relatively long occlusive are given in the following table:

| Word | Occlusive Point | Silence duration(ms) |
|---|---|---|
| वक्तव्य(vaktavya) | va-**k**tavya | 170 |
| मतदान(matdaan) | ma-**t**daan | 138 |

| | | |
|---|---|---|
| नेपाल(nepaal) | ne-**p**aal | 136 |

Table 1: Example words with occlusive

Following figures of wave files showing the occlusive silence illustrates our point.

Figure 1: Occlusive silence in the wave file for word "vaktavya"

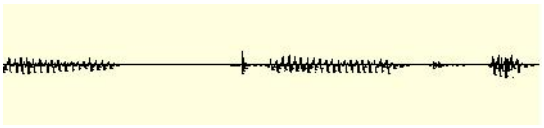

Figure 2: Occlusive silence in the wave file for the word "matadan"

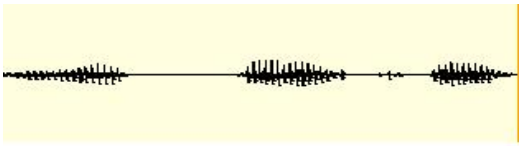

Figure 3: Occlusive silence in the wave file for the word "nepal"

In this analysis, we find two deciding factors 1) Minimum silence duration that was considered as pause by listeners and 2) Silence duration that can be distinguished clearly from occlusives. Based on these factors, we decided to consider silence duration of 200 ms as threshold to consider a pause mark.

Even for normal text, reading speech rate and duration of silence varies from speaker to speaker. Therefore, we tried to find more specific threshold for individual speaker based on the speech rate of speaker. We calculated the duration of each silence part present in our data. It was observed that there was very wide range of silence durations starting from 10 ms to 2000 ms present in the speech. We ignored the following:

- Silence durations less than 50 ms, which are generally not easily perceived.
- The longest 5% of silence durations, which are mostly exceptional cases, like 1042 ms, 1008 ms, 998 ms, 1111 ms.

Speech Rate (Syllable Rate), Arithmetic Mean (AM) and Geometric Mean (GM) of silence duration for all the 9 speakers is calculated (Table 1). From these results, no direct relation between GM and speech rate, and AM & speech rate can be established and nothing can be said about the minimum silence duration to consider as pause.

| Speaker List | Arithmetic Mean (AM) | Geometric Mean (GM) | Speech Rate |
|---|---|---|---|
| Speaker 1 | 141.24 | 92.6 | 3.88 |
| Speaker 2 | 131.87 | 98.2 | 4.59 |
| Speaker 3 | 103.98 | 100.73 | 4.67 |
| Speaker 4 | 135.51 | 105.93 | 4.31 |
| Speaker 5 | 127.51 | 107.26 | 4.73 |
| Speaker 6 | 119.06 | 109 | 5.1 |
| Speaker 7 | 112.22 | 114.42 | 4.27 |
| Speaker 8 | 137.17 | 115.12 | 3.74 |
| Speaker 9 | 159.85 | 128 | 3.54 |

Table: 2 AM, GM and Speech Rate

## 4 System Flow

The complete system flow for pause boundary detection and pause duration, takes speech data and corresponding text file as input and generates the following outputs:

1. Text file having content with proper pause marks
2. An error report giving hint about the doubtful places, which needs to be manually verified.

The system flow is depicted in the flow chart 1.

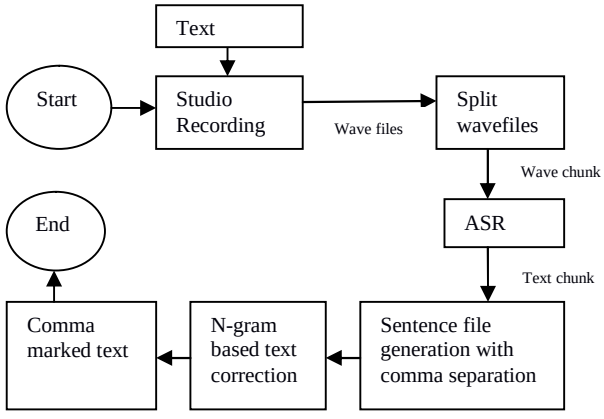


Figure 4: System flow chart

## 5 Results & analysis

We evaluated and analysed performance of the method in the following way.

### 5.1 Accuracy of text correction algorithm

As we mentioned, ASR output at sentence level is not perfect as it contains some addition, deletion, and change of words. Therefore, we apply the text correction algorithm and get significant improvement. The following table shows text improvement in two iterations. Correct sentence is the regenerated sentence, which is exactly same as original sentence (excluding pause marks).

| Language | Total No of Sentences | % Correct Sentence in ASR o/p | % Correct Sentences (Iteration1) | % Correct Sentences (Iteration 2) |
|---|---|---|---|---|
| Hindi | 2318 | 38 | 85.03 | 89.94 |
| Odia | 3570 | 49 | 81.45 | 88.01 |
| Marathi | 1889 | 45 | 89.35 | 93.06 |

Table 3: Text Correction Algorithm Accuracy

### 5.2 Accuracy of pause marking by the system

System has been tested with 4 sets of TTS training data:

1. Language: Marathi, Gender: Male
2. Language: Hindi, Gender: Male
3. Language: Hindi, Gender: Female
4. Language: Odia, Gender: Male

The TTS training data is available at IITM website(www.iitm.ac.in/donlab/tts/) and each set of data contains pause marked text file. System generated text file (with pause mark) is compared with manually pause marked text content. Given below is the result in Table 4:

| Language | True Positive | False Negative | False Positive |
|---|---|---|---|
| Marathi male | 95.51% | 1.44% | 3.02% |
| Hindi Male | 91.06% | 8.15% | 0.38% |
| Hindi Female | 97% | 1.21% | 1.3% |
| Odia Male | 97.06 | 0.02% | 0% |

Table 4: Results

True Positive: Manually labelled as pause and identified as pause by the system
False Negative: Manually labelled as pause and identified as not pause by the system
False positive: Manually labelled as not pause and identified as pause by the system.

True Negative is not calculated, as it is not meaningful.

We observed that the text correction method is unable to correct ASR outputs with missing text for certain speech utterances. For example in the following table, example 1 shows that word 'स्त्रीची' is completely missing from ASR output and so was not corrected as expected. Whereas, for other examples we can see that 'अंधार गेल्या' is has almost similar utterance as 'सुधारलेल्या' and so the sentence was corrected as expected. The same applies to other cases 3 and 4.

In the case of sentence 5 which is also not corrected as expected, the error can be traced back to the fact that if, the n-grams generated using the original sentence contain some gram (other than the expected) more near to the erroneous word, that 'other' gram will be used as replacement. We can see that, in case 5 'ही' should have been replaced by the 'ठीक' (at edit distance 2 from ही) but as there was 'मी' (at edit distance 1 from ही) which is closer to 'ही', the 'मी' was preferred which is a wrong choice. Mechanism to deal with such errors is being investigated.

| | |
|---|---|
| ASR output | 1. प्रतिष्ठा, फक्त वीर पत्नी, अथवा वीर माता होण्यात नाही, तर वीर स्त्री होण्यात आहे.<br>2. अंधार गेल्या संस्कृतीची माती हरवलेल्या खजिन्याचा धातु सारखा भास शतकांचे पाहा<br>3. ऊस स्पश सांग तुजला सांगुन काय गेला<br>4. मी ही दारावरच्या बेलने दचकून उठेल उंट च्या स्वागतासाठी<br>5. ही माझे कपडे मी आल्यावर धुवून टाकीन |
| Original Sentence | 1. स्त्रीची प्रतिष्ठा फक्त वीर पत्नी अथवा वीर माता होण्यात नाही तर वीर स्त्री होण्यात आहे.<br>2. सुधारलेल्या संस्कृतीची माती हरवलेल्या खजिन्याचा धातु सारखा भास शतकांचे पाप<br>3. तो स्पश सांग तुजला सांगुन काय गेला<br>4. तीही दारावरच्या बेलने दचकून उठेल तुमच्या स्वागतासाठी<br>5. ठीक माझे कपडे मी आल्यावर धुवून टाकीन |
| O/P of Correction Method | 1. प्रतिष्ठा, फक्त वीर पत्नी, अथवा वीर माता होण्यात नाही, तर वीर स्त्री होण्यात आहे.<br>2. सुधारलेल्या संस्कृतीची माती, हरवलेल्या खजिन्याचा धातु सारखा भास, शतकांचे पाप.<br>3. तो स्पश, सांग तुजला, सांगुन काय गेला.<br>4. तीही, दारावरच्या बेलने, दचकून उठेल, तुमच्या स्वागतासाठी.<br>5. मी, माझे कपडे, मी आल्यावर धुवून टाकीन. |

Table 5: Example outputs of Correction method

## 6 Conclusion

In this paper, we described the need of creating correctly pause marked text data with an automated process that can be used in the training phase of TTS building for achieving natural sounding high quality synthetic speech. We presented the mechanism to detect pause boundaries and duration of the pauses using ASR and customised text correction method. We also described methods to consider minimal pause duration that should work optimal. This would help those who would implement it for other languages. We described the experiments and results for Hindi, Marathi, and Odia, which looks promising. The system can readily be used for other languages and can contribute to further fine-tune and even achieve higher accuracy.

Currently, this system is marking only one type of pause. In future, we plan to work towards using multiple types of pause marks e.g., short pause, normal pause, and long pause. The basis of categorisation may be the duration of silence region. We also intend to study further to establish the relation between speech rate and silence duration threshold, which could be considered as pause mark. Along with the silence region, other factors like variation in intonation or energy may also be considered as the deciding factors for pause mark.

## 7 Acknowledgements

This work is carried out as a part of the research project “Development of Text to Speech Systems for Indian Languages with right pause mark Phase-II”, Ref. No. 11(7)/2011-HCC(TDIL), funded by Ministry of Electronics & Information Technology (MeitY), Govt. of India under TDIL Program. The authors would like to thank Prof Hema A Murthy from IIT Madras, Dr. Somnath Chandra and Ms. Swaran Lata from MeitY for their kind guidance and support.

## References

Anandaswarup Vadapalli et al., Significance of word-terminal syllables for prediction of phrase breaks in Text-to-Speech systems for Indian languages, in *Proceedings of 8th ISCA Speech Synthesis Workshop*, Barcelona, Spain, 2013, pp. 189-194

Andrew J Hunt and Alan W. Black. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In Proceedings of IEEE International Conference Acoustics, Speech, and Signal Processing, vol. 1: pp. 373–376.

Arun Baby et al., 2012. A Unified Parser for Developing Indian Language Text to Speech Synthesizers. *International Conference on Text, Speech, and Dialogue*,Springer International Publishing

Aswin Shanmugam Subramanian and Hema A. Murthy. 2014. Hybrid Approach to Segmentation of Speech Using Group Delay Processing and HMM Based Embedded Reestimation. *INTERSPEECH*

Atish Ghone et al., 2017.TBT (Toolkit to Build TTS): A High Performance Framework to Build Multiple Language HTS Voice *at Interspeech-2017 Conference, held at Stockholm, Sweden on August 20-24, 2017*

Daniel Povey et al . 2011. The Kaldi Speech Recognition Toolkit. in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, Hilton Waikoloa Village, Big Island, Hawaii, US.

Estelle Campione and Jean Véronis. 2002. A Large-Scale Multilingual Study of Silent Pause Duration. *Speech Prosody 2002.* Aix-en-Provence, France.

Heiga Zen et al., 2009. Statistical parametric speech synthesis. *Speech Communication*, vol. 51, pp. 1039–1064

Kishore Prahallad et al., 2006. Sub-phonetic Modeling for Capturing Pronunciation Variation in Conversational Speech Synthesis. in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Tolouse, France.

Krishnendu Ghosh and K. Sreenivasa Rao. 2012. Data-Driven Phrase Break Prediction for Bengali Text-to-Speech System. In *Contemporary Computing. IC3 2012. Communications in Computer and Information Science*, Vol 306. Springer, Berlin, Heidelberg.

Nicola Bertoldi and Mauro Cettolo. 2008. IRSTLM: An open source toolkit for handling large scale language models at Interspeech.

Parakrant Sarkar and K. Sreenivasa Rao. 2015. Data-driven pause prediction for synthesis of storytelling style speech based on discourse modes. *Electronics Computing and Communication Technologies (CONECCT) IEEE International Conference* pp. 1-5

Thi Thu Trang Nguyen. 2015. HMM-based Vietnamese Text-To-Speech: Prosodic Phrasing Modeling, Corpus Design System Design, and Evaluation. PhD thesis. Université Paris Sud - Paris XI

V. Kamakshi Prasad et al., 2004. Automatic segmentation of continuous speech using minimum phase group delay functions.

Vladimir*I.*Levenshtein,1966. "Binary codes capable of correcting deletions, insertions, and reversals."Soviet physics doklady. Vol. 10. No. 8.

http://www.festvox.org/download.html

https://www.iitm.ac.in/donlab/tts/database.php