

# Text-to-Speech Translation using Support Vector Machine, an approach to find a potential path for Human-Computer Speech Synthesizer

Rashmi S<sup>1</sup>

Department of Computer Science and Applications,  
Bangalore University  
Bangalore-560056, India  
rashmi.karthik123@bub.ernet.in

Hanumanthappa M<sup>2</sup>

Department of Computer Science and Applications,  
Bangalore University  
Bangalore-560056, India  
hanu6572@bub.ernet.in

Jyothi N M<sup>3</sup>

Department of Master of Computer Applications,  
Bapuji Institute of Engineering and Technology, Davangere-577004, India  
Jyothi\_nm@yahoo.com

**Abstract—** Text-to-Speech (TTS), an astounding feature to assemble computer with intelligence and to induce sound is seemingly a challenging task as it is related to the propagation of uncertainty with the input. This is because TTS evolves the input based on the probabilities and not with certainty ratios. TTS is accomplished by generating the sound structure/phoneme and then classifying these phonemes in the phonetic dictionary. The Wards' algorithms, BIRCH, Support Vector Machine (SVM) are used to figure out the appropriate sound representation for the given context. To distinguish correct elocution, the SVM procedures are equipped with the principles of pruning. The output was analyzed using divergent stages of uncertainty. In order to study the effect of the output 10 listeners were considered for determining Signal-to-Noise (SNR) ratio. SNR shows that the errors of both type phase and uncertainty were approximately 6% resulting 94% of accuracy. These results manifested that SVM stratagem can be used to obtain better results for TTS synthesizer.

**Index Terms—** BIRCH, Phonetic Analysis, Prosody, Support Vector Machine (SVM), Wards' Algorithm

## I. INTRODUCTION

A Text-to-Speech (TTS) synthesizer is a computer based framework that ought to have the capacity to peruse any content resoundingly, whether it was straightforwardly presented to the computer by a user or copied and pasted from any documents. Talking machines are not relatively new and started its inception in 18<sup>th</sup> century. But the problems at its root are not simple though. At first sight, this task looks not so difficult to achieve [1]. Truth be told, even the human's have the potential to efficiently pronounce any new words. They inherit this quality right from the childhood with the help of school teaching, principles of native language, watching television and so on. But the problem arises when the computer is involved. Hence a novel approach is required to address this problem. This task is very challenging and need lot of research to identify the correct pronunciation.

In spite of the current situation about the insights and the methods to resolve this complication, the advancements as of late achieved in the fields of linguistics and signal processing, we need to express few reservations. In reality reading words sounds very easy and it draws from the farthest profundities frequently unthought-of, of the human knowledge. To add to this, the ambiguous nature of the language worsens this problem.

There are several algorithms such as Support Vector Machine, Apriori algorithm, Hidden Markov Models that can be used for speech synthesizer systems. State-of-art systems possess high accuracy (>85%) rate and deploy Fourier transform to calculate the magnitude frequency speech segments [2]. Furthermore studies on speech recognition system reveal that the speech segmentation is a wide arena that is contained with various aspects of speech coding. In this research paper, an attempt to develop TTS interface by using support vector model is implemented. A study is also made to forecast the results by considering the human factor and the correct pronunciation produced by the system. The uncertainty ratio of the results obtained is compared against SNR phase conditions.

## II. PROBLEM STATEMENT, PRELIMINARY & RELATED WORK

Speech synthesis and phonetic analysis has become substantially robust over the past few decades and its quality and efficiency has improved considerably however there is a huge demand for the comparison of the techniques that has made this job simpler. In the early 1980's, a prominent number of speech synthesizers showed a reasonable performance perhaps there were no metrics to evaluate which one was the best. In this situation, a standard method for finding the error was discovered and it is called "Edit Distance/ Levenshtein distance. Edit distance compares two input strings and then identifies how many substitution, deletion and insertion are required in order to transform one string (reference string) to another string (hypothesis string) [2]. VODER (Voice Operated

recorDER) was invented by Homer Dudley in the year 1939. This had a keyboard console which was administered by 10 parallel pass filters. Random noise generator aided for the production of unvoiced sounds and the relaxation generator for voiced sound [3].

In the mid 2000, speech recognition systems were taken to a new horizon and served as a platform for many research activities. By 2005, there was speech recognition systems developed for three languages (English, Spanish, and Mandarin). This included five voices at the speed of 10 Hertz with 96 kHz/24 bit recording precision [4]. Human perception of sound differs from the computer generated sound. Many researchers have worked on the phase segmentation. The authors have shown the minimal usage of data to constitute the distortion of frequency and threshold [5]. In another work the focus was on the consonants and the frequency range. It is clear that the sound perceived by human varies as this difference is just notable when compared to the human speech spectrum [6].

### III. ARCHITECTURE OF TTS

This section describes the proposed architecture of TTS. The implementation of TTS is combined with the techniques of data mining in order to make it more prevailing and strong. Data mining aims at retrieving useful patterns from a huge repository that is rather archived and not being used. The discovery of such useful patterns is required for business decisions, knowledge processing, and analytics.

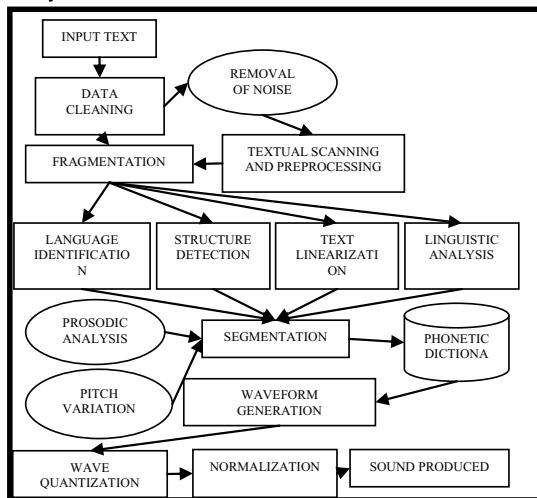


Fig. 1- Architecture of TTS Speech Synthesizer,  
Transformation of Grapheme to Phoneme

### IV. IMPLEMENTATION OF THE PROPOSED SYSTEM

As fore mentioned though TTS system looks as an easy task there are various complications included. When text is converted into speech it is of utmost important to obtain

good accuracy otherwise it turns obsolete. Therefore speech synthesizer should match that of humans or even better. Hence, our proposed architecture states an experimental limit on the accuracy pace.

#### A. Data cleaning

The text provided as input contains multiple noises such as special characters, multiple white spaces, numeric characters. When the data is forecasted, these noises become outliers. Therefore removing such outliers becomes the first step towards building Speech synthesizers. To do so, Wards' method of cluster analysis is used. This is an agglomerative clustering approach. Wards' method begins with n-clusters and moves out to form 1-cluster (figure.2). The data in each cluster will have numerous multivariate variables. When scatter plotted the cluster points results in elliptical shape (figure.3). The multivariate and the elliptical shape will result in n X n dimension graph. Let  $P_{xyz}$  indicate the value for each variable in observations held by one cluster 'i'. For every cluster there is an uncertainty measure perceived as error. The sum of errors is given by the equation mentioned in (2) and (3)

$$\text{Sum\_of\_Error, } SE = \sum_x \sum_y \sum_z |P_{xyz} - \bar{P}_{x,y}| + \varepsilon \quad (2)$$

Equation (2) indicates the cost of formation of a single cluster 'i'.  $\varepsilon$  is the error.  $\varepsilon$  is the cost of comparing every word in a cluster. Here computation of each observation for every variable is made against the total value of the cluster. If the value of SE is small and negligible it means that the data are near to the cluster.

$$\text{Sum\_of\_SE, SSE} = \sum_x^k |SE|^2 + \int_y^k \varepsilon * \eta \quad (3)$$

In a given text there can be 'n' cluster. SE indicates the error of 1-cluster as shown in equation (2). The overall computation for 'n' cluster is shown in equation (3).

This results in outlier's formation. For example, let us consider the input text of the form, "I am William Wallace; I am in 5th grade and aged 10 years. I live in Amsterdam-Netherlands". Equation (2) and (3) are combined to show the formation of 3-clusters for the above example. This is illustrated in figure.2. There are certain points which can be neither classified as criterion defined nor based on the certainty measures such as resemblance, likeness hence those are formed as a new cluster/outlier. This results in two branches partial and full interpretation of complete linkage in cluster trees. As a result a dendrogram is formed. This is summarized in the figure.3.

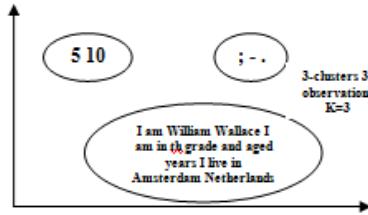


Fig. 2- Formation of Clusters using Wards' Cluster Analysis

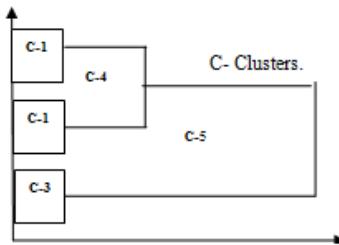


Figure 3- Wards' Cluster

#### B. Language identification

Language model is huge consortium to think of. In the proposed speech synthesizer the text-to-speech conversion happens only for English language. Therefore in this step a new approach is put forward to identify the given language as English or not. This is done by using one of the famous Data Mining techniques called Classification.

Classification is a process of assigning an object to an existing class based on the properties and behavior of the object [7]. In order to identify the language, we use Rule-based classification model. This model works by generating the set of rules upon which the classification acts. The stop words can be used for language recognition. Stop words are the basic elementary list of data that uniquely identifies a language or a dialect. A list of such stop words is prepared before hand for English language. In the proposed architecture the list of stop words are treated as training data. The input text is compared with the training set. The training set is later mapped on to the If-Then rules of the classifier. Based on the output, these rules correctly classify the language. The current set up of the proposed system works only for the English language but can be extended for other languages by constructing the stop word list of the respective language. Table 1 describes the RBC algorithm that is proposed which correctly identifies English language. This is a rule based classifier which is constructed by extracting If-Then rules.

Table 1- RBC Algorithm

<pre> Algorithm: RBC Input→ S, the input text given by the user           D, a data set of stop words           Atri_val, the set of all attributes and their associated           values Output→ Language is identified as English Rule_Set= { } Here Rule_Set is initialized to empty indicating that the rules learned so far is empty Count=size of the sentence in terms of word count For each "word" of S do Repeat Rule 1 If &lt;word&gt; = &lt;D&gt; Then add "word" to atrival If atrival=count Then language=English Else Do not claim </pre>
--

#### C. Text linearization and linguistic analysis through phonetic analysis

So far the formation of cluster was shown. In this module the formation of phonetic dictionary is discussed which consists of phonetic transcription that targets on the way a word should be pronounced. Though the task of converting the grapheme looks simple as it could be achieved by constructing word-reference look-up table the real challenge is with the pronunciation and today all most all the dictionaries contain phonetic transcription

A dictionary of phonetic words and its related transcription is built. This list contains a small sub set of word set. A sample of phonetic dictionary was constructed. In the first step, the phonetic transcription a sample of 250 words and 250 sentences were added in the dictionary. The reason behind this is that small set of words could be used as a training sample for phonetic dictionary for an untrained word sample.

The implementation of phonetic dictionary can be achieved in two ways. Those are I) Dictionary method and II) Rule based Support vector machine

##### I) Dictionary based

In this approach a dictionary is built which contains the phonetic transcription of all the words in the “Universe”. This method could lead to numerous lists of words and it is highly time consuming. The other drawback of this method is that when a text is misspelled the dictionary will not be able to produce the corresponding phonemes. For example: “what” and “wat” or “come or cum or com”. Though the pronunciations of these words are similar the interface will not be able to produce the result.

##### II) Rule Based Classifier

According to Hunnicut 80, 2000 words are typically enough to span around the 70% of words in English. So in order to achieve stable TTS system, a phonetic dictionary of around 1000 words is constructed. When a new text appears which do not exist in the dictionary then one of the classification techniques called Support Vector Machine (SVM) is used. In the upcoming section SVM is explained in detail.

##### Support vector machine:

From the explorative analysis and through intense literature survey on various type of data mining techniques it was found that the SVM is suitable for inscribing the phonetic model. SVM was developed by Vladimir Vapnik and it is a supervised learning method under classification and regression under data mining. This is one of the promising learning algorithms which yield effective results. SVM is based on statistical learning theory.

SVM performs the classification of class models by defining a hyper plane. SVM is expounded by two types of classifiers. One is “Linear Separable” and the other is “Perceptron”. In order to learn a classifier the following analogy is applied.

Set of training data is given by the x and y coordinates ( $X_i, Y_i$ ) where  $i=1,2,3,4,\dots,N$ .

$X_i \in Ed$  where  $Ed=$  Edit distance value

$Y_i \in -1, +1$  where  $Y_i$  = the range of possible attempts in order to produce the corresponding phonemes for a given text/sentence.

$f(x)$  is a classifier such that,

$$f(x) = \begin{cases} >= 0 \text{ then } Y_i \text{ ranges only on the positive side of the coordinates} \\ <0 \text{ then } Y_i \text{ ranges on the negative axis of the coordinates} \end{cases}$$

Therefore,  $f(x) \geq 0$  stands for correct classification

Then according to the theory of SVM, if such correct classification happens then it is said to be linearly separable otherwise perceptron. Equation (6) provides a set of training data which are used to produce the phonemes.

$$f(x) = E(Wt) X + e \quad (6)$$

where  $e$ = Signal error,  $E$ =Edit Distance,  $Wt$ = Weight factor which is given by the number of points that falls on the hyper plane given as inputs,  $X$ = Range of value

Furthermore, a line is said to be bad if it crosses too close to the points as it will be noise sensitive and the corresponding characters to phonemes generation are neglected and hence the goal to is draw a line as far as possible from all the points but can be linearly separable. This line should be drawn in such a way that it should have the same distance between two different set of training data. Training samples that are near and close to this line are called support vectors. Observe the figure.6. It shows the formation of support vectors and correct phonetic transcription for an input text "beautiful" – "bjutəfel". Every character in the context appears as vector points on the hyper plane. The phonetic transcription that is stored in the phonetic dictionary is retrieved for every character and the new words fall on the separation line and forms as support vectors and finally the phonetic transcription is rendered as an output.

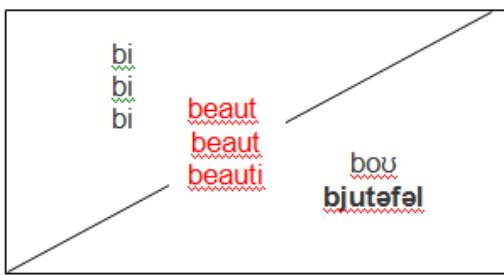


Fig. 4- Formation of Support vectors and the related cluster points.

#### IV. RESULT AND EVALUATION

The results of all the three phases are shown in the interface developed that meets and satisfies the methodologies described in this paper is shown in the below figure.6. This figure indicates the overall development of phonetic language processor. This is developed using .net platform. This module consists of 4 phases 1) Language Detector, identifies the language of the input as described in the 4.2 2) Audio to Text. This phase deals with the conversion of Speech-to-text. The mechanism of implementing this feature is beyond the scope of

this paper 3) Text-to-Speech. This phase demonstrates all the techniques explained so far in this paper. 4) Grammar check. As the name indicates this is used to check the syntactic structure of a language. This phase is again beyond the scope of this paper. The output shown in the figure is discussed in the 2<sup>nd</sup> phase of this processor. A dictionary of about 500 words was built. The dictionary serves as database which contains long words such as "Apocalyptic" and on the other side it contains some of the confusing pairs such as four/fourth, a.m/am, what's/what, what's/was. The letter 'a' and 'the' has different forms of pronunciation.

Therefore the primary task was to recognize all such words that are difficult and eloquent in nature however these words are pronounced wrongly by the user or in some situation it has to be pronounced differently. Henceforth the phonetic dictionary is built for all these words and this will serve as the training set. Along with the phonetic transcription of word to word model, around 100 sentences were chosen randomly from internet and then later the speaking pattern was manually assigned to all these sentences. For each of the 100 sentences, sentence model was constructed by using the word-pair that are contrive in the phonetic dictionary. Given a text/sentences, the existing context in the phonetic dictionary were directly picked however for the new words that are present in this corpus, SVM model of generating the pronunciation is used. It must be noted that the training/testing sets was constructed based on one universal language English. After using the SVM model the sentence model was tagged into phone (phonemes) model. At this stage we have the phonetic transcription for the entire input context.



Fig 5- TTS Speech Synthesizer

#### V. PERFORMANCE ANALYSIS

Performance analysis for any experiment is very important as this evaluates the correctness of the results achieved. For preprocess of the values in the given input text, considering 'i' iterations for evaluating different algorithms as indicated in the earlier section it is evident that word segment is classified into various phonemes structure and further evaluated for noise removal, textual scanning and analysis. This estimation is given by,

$$\stackrel{\Lambda}{Y}_{i+1}(n) = \frac{\sum_{i=1}^{\infty} (1-\eta) \stackrel{\Lambda}{Y}(\omega) \int_{-\pi}^{\pi} \stackrel{\Lambda}{Y}_{i+1}(n) \int_y^k e * \eta}{\prod_{\pi} h^2(i * e^{1/2})} --(9)$$

The variables in the equation (9) indicate the same meaning as it indicated so far in other equations however  $e$  represents the time taken by the synthesizer to arrive at the output for the given input text. It was found that for larger input size the value of  $e$  was also found to be higher and this result in a small variation of the total time delay and on an average 4% was the maximum increase in the overall efficiency rate. This sifted to 0% for the minimum values of SNR. Equation (9) shows the calculations of the error encountered during the training phase along with the complexity error which increases as there is increase in the corpus. Our goal is to i) Maximize margin so that we get the support vectors of these words that are already in the training set. ii) Minimize training error. Given a stable amount of training data, it is feasible to find the attainable training model. With this technique, any new sentences will be pronounced with good precision. To further analyze the results the obtained it was tested using F-measure. Recall and a precision are the efficiency measure to prove the correctness of the proposed system.

RECALL is defined as the ratio with the number of correct/relevant records that is retrieved to the actual total of the relevant record.

This is expressed in terms of percentage and given by the formula  $(A/(A+B)) * 100$

PRECISION is defined as the ratio with the number of correct/relevant records that is retrieved to the actual total number of irrelevant records that are retrieved.

This is expressed in terms of percentage and given by the formula  $(A/(A+C)) * 100$

Using the designations above:

- A = Total number of relevant data/relevant retrieved,
- B = Total number of relevant records not retrieved, and
- C = Total number of irrelevant records retrieved.

For sentence

Total sentence taken for test:

2500 sentence

Pronunciation produced : 2472

Out of 2472 retrieved, 2360 were relevant

A:2360

B:2500-2360=140

C:2472-2360=112

**Recall:  $(A/(A+B)) * 100 = 94\%$**

**Precision:**

$(A/(A+C)) * 100 = 95\%$

For words

Total words taken for test: 1200

words

Pronunciation produced : 1129

Out of 1129 retrieved, 1098 were relevant

A:1098

B:1200-1098=102

C: 1129-1098=31

**Recall:  $(A/(A+B)) * 100 = 91\%$**

**Precision:**

$(A/(A+C)) * 100 = 97\%$

## VI. CONCLUSION

Data mining is a decision support model that aids for the pattern identification in huge corpuses. SVM model under classification technique of data mining is used to design the

Text-to-Speech interface. Text-to-Speech (TTS) is a very challenging task. This paper proposes TTS interface along with the complexion and milestones related to components of TTS system for English language. A novel approach is proposed using SVM, Wards' and BIRCH model. Phonetic dictionary was built for a small set of words and sentences. This is considered as a training set where a new word is tagged on the existing dictionary using SVM model of classification in Data Mining. Furthermore pronunciation model was studied for different features of linguistic analysis such as textual scanning & pre processing, phonetic analysis and prosodic analysis. The SNR conditions and phase distribution were studied for a diverse range of inputs. It was seen that phase distribution across the error rate and SNR values increases with the increase in phase noise. The results achieved are encouraging and the average of the results obtained was found to be 96% efficient. Further study on TTS synthesizer can be made for other languages using the proposed methodologies

## REFERENCES

- [1] Ameera Al-Rehili et al, *International Journal of Science and Applied Information Technology*, Vol.1, No.2, May-June 2012, ISSN No. 2278-3083
- [2] M. Hanumanthappa et al, *2015 International Conference on Computer Communication and Informatics (ICCCI -2015)*, Jan. 08 – 10, 2015
- [3] Piyush Mishra et al, *International Journal of Computer Applications*, Vol.70, No.26, May 2013, ISSN No. 0975-8887
- [4] D. Sasirekha et al, *International Journal of Soft Computing and Engineering*, Vol.2, No.1, March 2012, ISSN No. 2231-2307
- [5] Guangji Shi et al, "On the Importance of Phase in Human Speech Recognition", *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, VOL. 14, NO. 5, SEPTEMBER 2006
- [6] Yugal Kumar et al , "A New Initialization Method to Originate Initial Cluster Centers for K-Means Algorithm ", *International Journal of Advanced Science and Technology* Vol.62, (2014), pp.43-54, *IEEE conference*.
- [7] Timothy Baldwin et al, "Association for Computational Linguistics", the 2010 Annual Conference of the North American Chapter of the ACL, Pages 229-237, June 2010.