Concatenative Synthesis of Persian Language Based on Word, Diphone and Triphone Databases

Reza Javidan (Corresponding author)
Computer Engineering Department
Islamic Azad University – Beyza Branch, Fars, Iran
Tel: 98-917-315-9656 E-mail: reza.javidan@gmail.com

Iman Rasekh
Computer Engineering Department
Islamic Azad University – Arak Branch, Arak, Iran
Tel: 98-917-3072-2481 E-mail: iman.rasekh@gmail.com

Abstract

In this paper a new Persian text-to-speech system based on concatenative speech synthesis approach is proposed. Nowadays, concatenative method is used in most modern TTS systems to produce artificial speech. In concatenative method, selecting an appropriate unit for creating a database is a challenging task. In the proposed approach, such database is created with different sizes of speech units and is used to produce speak utterances which include words, diphones and triphones. For the synthesis process, Diphones and triphones which are smaller speech units are used to achieve unlimited vocabulary of speech, while the word units are used for synthesizing which make limited set of sentences. Moreover, a dictionary of 600 common Persian words is built. The simulation results on prototype data show the effectiveness of the proposed method.

Keywords: Persian text-to-speech synthesis, Artificial neural networks, Concatenative synthesis, Word, Diphone, Triphone

1. Introduction

A Text-To-Speech synthesizer (TTS) is a computer-based program in which the text is processed by a computer and the computer reads the text aloud. For most applications, there is a demand on the technology to deliver good and acceptable quality of speech. High quality speech synthesis in electronic text format has been a focus of research activities in past two decades, and it has led to an increasing horizon of applications (T. Dutoit. 1997). To mention a few, commercial telephone response systems, natural language computer interfaces, reading machines for blind people and other aids for the handicapped, language learning systems, multimedia applications, talking books and toys are among the many examples.

The speech synthesizer consists of two main components, namely: Text processing component and Digital Signal Processing (DSP) module. The text processing component has two major steps: *Step1:* Converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words; this process is often called text normalization. *Step2:* It converts the text into some other representation and output it to the DSP module or synthesizer, which transforms the symbolic information it receives into speech (Kain Alexander, and P.H. Santen Jan Van. 2003).

The primary technologies for generating synthetic speech waveforms are including formant synthesis and concatenative synthesis (R. J. Deller, et al. 2000). Each of these technologies has its own strengths and weaknesses and the intended users of a synthesis system will typically determine which approach will be used. Formant synthesizers, which are usually controlled by rules, have the advantage of having small footprints at the expense of the quality and naturalness of the synthesized speech (Z. Namnabat, and M. M. Homayunpoor. 2004.). Speech synthesizer that is built as a result of this article depends on the concatenative synthesis approach. In concatenative synthesis the waveforms are created by concatenating parts of natural speech recorded by humans. The easiest way to produce intelligible and natural synthetic speech is to concatenate prerecorded utterances. This method is limited to one speaker and one voice and the recorded utterances require a larger storing capacity compared to the other methods of speech synthesis (O. Karaali, G. Corrigan and I. Gerson. 1996).

Various researches have been done to synthesize speech by different means and for different languages. In 1987, Sejnowsky and Rosenberg (F. Daneshfar, W. Barkhoda, and B. ZahirAzami. 2009) constructed a neural network that learned to pronounce English text, the system called NET Talk. It was built using a large number of parallel network systems that can capture a significant number of the regularities and many of the irregularities in English pronunciation to convert strings of the English text into strings of phonemes. In addition the research is extended to other methods than NET Talk for example; Karaali et al. (M. Hosaini, and M. Homayonpour. 1999) constructed a rule-based system that uses two neural networks. The first one is a Time-Delay Neural Network to convert a phonetic representation of speech into an acoustic representation and then into speech. The other one is used to control the timing of the output speech. Regarding Persian speech synthesis, Nasirzadeh et al. (T.J. Sejnowski, and C.R. Rosenberg. 1987) proposed the first Persian concatenative text-to- speech synthesis system that uses diphone/sub-syllable method to construct the spoken utterances. The speech units they used are chosen where the co-articulation effect of the classical Persian is minimal. They also proposed extension of the set of speech units to improve the quality of the output speech (Z. Namnabat, and M. Homayunpoor. 2004).

In the proposed method in this article, the recorded utterances are divided into smaller speech units, such as: words, syllables, phonemes, diphones and sometimes triphones. Word is the most natural unit for the written text and suitable for systems with very limited vocabulary. Diphones are two adjacent halfphones (context-dependent phoneme realizations), cut in the middle and joined into one unit. Triphones are like diphones, but contain one phoneme between steady-state points (half phoneme-phoneme-half phoneme). In other words, a triphone is a phoneme with a specific left and right context (H. R. Abutalebi, and M. Bijankhan. 2000).

The rest of the paper is organized as follows: Section 2 introduces the Persian TTS General Architecture. In Section 3 three different Methods for designing our database are presented; word, diphone and triphone based systems are designed. Comparison between these systems and experimental results are presented in Section 4. Finally, Conclusions and remarks are explained in Section 5.

2. Persian TTS General Architecture

The general architecture of a Persian Text-To-Speech system is shown in Figure 1. The input to the system is the result of queering an existing search engine which is capable of retrieving Persian textual data. The text-to-speech synthesis procedure consists of two main phases (F. Hendessi, A. Ghayoori, and T. A. Gulliver. 2005):

a) The first phase is text analysis. In this step the input text is pre-processed and then classified using artificial neural networks, we used unsupervised learning paradigm, specifically the Kohonen learning rule. Such a network can learn to detect the features of the input vector.

b) The second phase is the generation of speech waveforms. In this research, concatenative speech synthesis approach is used .The Post processing unit is designed to realize concatenative synthesis. This unit is used to smooth the transitions between the concatenated diphoneses. The development of a high quality TTS system needs an appropriate database of speech units. Diphoneses are the main speech units used during the course of this study. The used Persian diphonese database was prepared at “Amir Kabir University of Technology”, this database contains 218 speech units (W. Barkhoda, F. Daneshfar, and B. ZahirAzami. 2008).

The detail of above architecture is explained in Subsections 2.1 to 2.3

2.1 General Architecture Main Parts

2.1.1 Text pre-processing (text normalization)

Before the words enter the neural network, a series of preliminary processing has to be fulfilled.

a) The punctuation marks are removed.

b) The numbers are identified. For example: 123 → “صد و بیست و سه” (W. M. Thackston, and Sorani. 2006).

c) The Abbreviations and Acronyms are expanded into full words (A. Rokhzadi. 2002) as shown in Table1.

d) Check the interaction of stress and intensity (A. Rokhzadi. 2002), (M. Kaveh. 2000) as shown in Table2.

e) The final step is to prepare the words as input vectors for the neural network.

In the next Subsection, a neural network has been used to extract allophones.

2.1.2 The Neural Network Architecture

Neural Networks can learn from a database and can recognize allophones properly. In this research four sets of neurons in the input layer is employed, each having 35 neurons for detection of the 35 mentioned standard symbols. A sliding window of width four provides input phonemes for the network input layer. Each set of input layer is responsible for one of the phonemes in the window. The aim is to recognize the relevant allophone to the second phoneme of the window (S. Baban. 2005), (R. J. Deller, and et al. 2000). The output layer has 66 neurons (corresponding to the 36 Persian Diaphones which is used here) for the recognition of the corresponding allophones and the middle layer is responsible for detecting language rules and it has 60 neurons (these values are obtained empirically). The neural network accuracy rate is equal to 98 %. In standard script, 35 standard symbols are spotted. Notice that this is more than the number of Persian phonemes, because three standard symbols for space, comma and dot are also included.

Neural networks only recognizes numerical inputs, therefore, the ASCII code of each character is taken and replaced with its corresponding binary representation. Next, the 0's were replaced with (-1)'s to discriminate them from trailing zeros that will be added later. Now the text is ready to be processed and classified by the neural network.(M. N. Rao, S. Thomas, T. Nagarajan, and H. A. Murthy. 2005). Figure 2 shows the main architecture of the neural network.

2.2 Text to speech (TTS) conversion

When building a speech synthesizer, one has to decide which synthesis unit should be chosen. There are different unit sizes and each choice has its own advantages and disadvantages. The longer the unit the more accuracy you get, but at the expense of the number of data needed. In this research we created three models to handle different sizes of units. These units are words, diphones and triphones (M. Goldstein. 1995).

2.3 Text post-processing (interpolation)

In the post processing unit three interpolation methods are used to smooth the transitions between speech units, namely: Linear interpolation, Spline interpolation and Cubic interpolation. When applying interpolation on the output speech, the results showed that the linear interpolation made no changes on the signal. Meanwhile the Spline interpolation has an effect but it's not the desired one since this kind of interpolation caused the signal to oscillate. The Cubic interpolation could successfully smooth the transitions between diphones, but it had a slight effect in improving the quality of the speech when it was played (J. Logan, B. Greene, and D. Pisoni. 1989).

3. The Proposed Methods

3.1 The word Based TTS System

Systems that simply concatenate isolated words or parts of sentences are only applicable when a limited vocabulary is required (typically a few hundreds of words). The sentences which are going to be pronounced follow a very restricted structure (R. J. Deller, et al. 2000). In this model, a dictionary containing 600 words that are commonly used in Persian is built. The goal of this procedure is to generate the corresponding speech of each word in the dictionary. Since our database of speech doesn't contain complete words, we constructed each word out of its diphone sequence.

3.1.1 Training the words of the Database

To train the words of the dictionary, each word is converted into its diphone sequence then passed to the pre-processing unit as explained previously. Neural networks require that all inputs are of the same length, so we chose a vector length of 164 with regards to longest word in the dictionary. Thus, words producing a vector shorter than 164 are padded with trailing zeros. Figure 3 shows the functional diagram of the training process, the input feature vector is passed to the network at the beginning. The neural network in turn produces a cluster representing the input. Each cluster is then passed to the converter module and is converted into a pattern of 1's and 0's for comparison purposes to be performed later. Now, the pattern is mapped to its corresponding speech signals and saved in a look-up table. This process is performed for all the words of the dictionary.

3.1.2 Synthesizing words

In this process the input text is tokenized into single words and each word is processed individually. Each word goes through the same training process to produce the feature vector and the output pattern. This pattern is then compared with the patterns in look-up table and classified by the Euclidean distance metric. At last, the recognized word is mapped to the corresponding sound and output as a speech. The synthesis procedure is shown in Figure 4 (M. Malcangi, and D. Frontini. 2009). The Euclidean distance exceeds a certain threshold; this means that the word hasn't been recognized as one of the trained words. In this case, it will be added along with its corresponding speech to the look-up table.

3.2 The diphone Based TTS System

To create a more flexible model which adapts to any new input data, for unrestricted speech synthesis we need to use shorter pieces of speech signal, such as diphones and triphones. The concept of this model is to use the diphones as the speech synthesis units (H. Sak. 2004).

3.2.1 Training the diphones of the Database

This step aims to generate a mapping between the textual diphones and their equivalent speech units. Each diphone is represented by two characters, consequently producing a vector of 14 elements. The training process is similar to that of the word model, except that the produced pattern is mapped to the equivalent speech unit of that diphone. This process is repeated for all the diphones in the database. The training process is shown in Figure 5 for the speech. In this figure, the words are automatically broken down to their diphone sequence. Each diphone will be converted into a feature vector then trained by the network to finally produce the pattern. This pattern is classified by the Euclidean distance and the corresponding diphone speech is fetched. This process is repeated for all the diphones. The output diphone units are saved in a speech buffer until text reading is finished. After that, the speech segments are concatenated together to produce a spoken utterance as shown in Figure 6.

3.3 The Triphone Based TTS System

This model uses longer segmental units (triphones) in attempt to decrease the density of concatenation points, therefore provides a better speech quality. The diphones in the speech database were used to build a database of 300 triphones, each triphone is built up by concatenating two diphones. For example, the triphone ‘Dit’ consists of the diphones ‘Di’ and ‘it’ connected together. Just note that we built triphones this way provided that the speech units in our hands are diphones, but this is not how triphones are actually constructed (H. Sak. 2004).

3.3.1 Training the triphones of the Database

This procedure is the same as the one used to train diphones, with a difference of the size of the input and output units. A triphone is presented by three characters producing a feature vector of 21 elements. When the pattern is generated, it’s mapped to the equivalent triphone speech unit. This process is repeated until the whole 300 triphones are trained.

3.3.2 Synthesis using triphones

To generate spoken utterances in this model, the words are automatically segmented into triphones. These triphones are converted into feature vectors of 21 elements and they go through the same procedure as the diphones (shown in Figure 7).

4. Experimantal Results and Discussion

The proposed system was built and evaluated using the Matlab7 programming language. To evaluate the accuracy of the synthesizer, different sets of sentences and words are input to the three models (word, diphone and triphone). In order to evaluate the quality of the system, a subjective listening test was conducted. The test sets were played to eight volunteer listeners (4 females and 4 males), which their ages range from 20-34 years. All the listeners are native Persian (6 of 8) and Afghanian (2 of 8) speakers and have no experience in listening to synthesized speech. The speech was played by loudspeakers in a quiet room and each listener was tested individually. For evaluating our proposed TTS systems and comparing them, various tests have been carried out.

In the first test, a set of seven sentences which were produced with each system was used as the test material. The test sets were played to volunteer listeners. The listeners were asked to rate the systems’ naturalness and overall voice quality on a scale of 1 (bad) to 5 (good). The volunteers didn't know anything about the sentences before listening to them. The obtained test results are shown in Table 3.

To determine the system's intelligibility, a second test has been carried out. In this test, the listeners were asked to write down the text they understood; then WCR (Word Correctness Rate) and SCR (Syllable Correctness Rate) were computed using the following equations (H. Sak. 2004):

$$WCR = \frac{Correct\ Words\ Number}{Toral\ Words\ Number} \tag{1}$$

$$SCR = \frac{CorrectSay\ llablesNum\ ber}{TotalSylla\ blesNumber} \tag{2}$$

Table 4 shows the results for various systems. According to these results, all systems' intelligibilities (especially that of the triphone based system) are acceptable. The recognition accuracy of the ten sentences and twelve independent words in diphone and triphone systems based on SCR Ratio are shown in Figure 8 and Figure 9.

In this research, we design a Text-To-Speech system for synthesizing retrieved Persian text from different resources, specially the Internet. We use the neural networks with unsupervised learning paradigm, which is proved to be a good tool for text to speech synthesis. As Compared to formant synthesis and other technologies, concatenative synthesizers produce more natural speech, but they are usually limited to one speaker and one voice and usually require more memory capacity than other methods. The experiments are done over the three models. The system that we build indicates that words are accurate and fast choice for synthesis, but they require large storage space and only useful for limited vocabulary applications. Moreover in this application, diphones make the best flexibility in building voices over words and triphones, since they produced good quality voice with small number of units compared to the other two models. The process of concatenating speech units for synthesis causes many cuts in the speech signal, which reinforces the importance of performing some, post-processing to enhance the quality of speech. Among the interpolation methods explained in this study, linear interpolation had no effect in alleviating the sharp transitions, so a smoother interpolating function is desirable. Splines are smooth interpolants but didn't produce the desired improvement in our work, since they caused the signal to oscillate. The cubic interpolation showed better performance compared to linear and spline interpolations. On the other hand, cubic interpolation caused a slight improvement in the speech quality, due to the very short period we performed the interpolation on.

5. Conclusion

In this research, a Persian text-to-speech synthesis system is proposed. Artificial neural network with unsupervised learning paradigm is used to build the system and different types of speech units. Moreover, the model is used to synthesize the desired utterances which are: words, diphones and triphones. The experimental results over the system showed its ability to produce unlimited number of words with high quality voice and high accuracy in converting the written text into speech. The resulting model obtained accuracy by the word and diphone models was 99% and by the triphone model was 86.5%.

The most important challenge in concatenative TTS is choosing appropriate unit for the database. This unit must warranty smoothness and high quality speech. Moreover, creating the database must be reasonable and inexpensive. For example, Word, diphone and triphone, are usually considered as appropriate for all-purpose systems.

Unit selection method can produce high quality and natural output speech. Developing a TTS system using unit selection and combining it with other methods is our goal in the future works.

References

A. Rokhzadi. (2002). Persian Phonetics and Grammar. *Tarfarnd press*, Tehran, Iran.

F. Daneshfar, W. Barkhoda, and B. ZahirAzami. (2009). Implementation of a Text-to-Speech System for Persian Language'. *ICDT'09*, Colmar, France.

F. Hendessi, A. Ghayoori, and T. A. Gulliver. (2005). A Speech Synthesizer for Persian Text Using a Neural Network with a Smooth Ergodic HMM. *ACM ransactions on Asian Language Information Processing (TALIP).*

H. R. Abutalebi and M. Bijankhan. (2000). Implementation of a Text-to-Speech System for Farsi Language. *Sixth International Conferenceon Spoken anguage Processing (ISCA).*

H. Sak. (2004). A Corpuse-Based Concatenative Speech Synthesis System for Turkish, *M.Sc. Thesis*, Bogazici University.

J. Logan, B. Greene, and D. Pisoni. (1989). Segmental Intelligibility of Synthetic Speech Produced by Rule. *Journal of the Acoustical Society of America*, JASA vol. 86 (2): 566-581.

Kain Alexander, and P.H. Santen Jan Van. (2003). A speech model of acoustic inventories based on asynchronous interpolation. *Proceeding of the EUROSPEECH-2003*, USA., pp: 329-332.

M. Goldstein. (1995). Classification of Methods Used for Assessment of Text-to-Speech Systems According to the Demands Placed on the Listener. *Speech communication*, vol. 16: 225-244.

M. Hosaini, M. Homayonpour. (1999). Farsi Text-to-Phoneme Conversion Applying Neural Networks. *8th Iranian Conference on Electrical Engineering*, vol 1, pp 195-163.

M. Kaveh. (2000). *Persian Linguistic and Grammar (Saqizi accent)*. Ehsan Press, first edition, Tehran, ISBN 964-356-355-3.

M. Malcangi and D. Frontini. (2009). Implementation of Three Text to Speech Systems for Kurdish Language. *Springer Berlin / Heidelberg*, Vol. 5856/2009, ISSN: 0302-9743.

M. Malcangi D. Frontini. (2009). Language-independent, neural network-based, text-to-phones conversion . *Springer Berlin / Heidelberg*, Volume 73, Issue 1-3, Pages 87-96, ISSN:0925-2312

M. N. Rao, S. Thomas, T. Nagarajan, and H. A. Murthy. (2005). Text-to-Speech Synthesis using syllable-like units. *National Conference on Communication*, India.

O. Karaali, G. Corrigan and I. Gerson. (1996). Speech synthesis with neural networks. *Proceeding of the World Congress on Neural Networks*, San Diego, pp: 45-50.

R. J. Deller, and et al. (2000). *Discrete time processing of speech signals*, John Wiley and Sons.

S. Baban. (2005). *Phonology and Syllabication in Persian Language*. Persian Academy Press, first edition.

T. Dutoit. (1997). High-quality text-to-speech synthesis: An overview. *J. Electron. Eng. Special Iss. Speech Recog. Synthes*., 17: 25-37.

T.J. Sejnowski and C.R. Rosenberg. (1987). Parallel networks that learn to pronounce English text. *Complex Syst*., 1: 145-168.

W. Barkhoda, F. Daneshfar, and B. ZahirAzami. (2008). Design and Implementation of a Persian TTS System Based on Allophones Using Neural Network. *SCEE'08*, Zanjan, Iran.

W. M. Thackston, Sorani. (2006). A Reference Grammar with Selected Reading. *Iranian Studies at Harvard University ,* Harvard.

Y. Samare. (1984). *Farsi Language Phonology.* Tehran University Press ,Tehran, Iran.

Z. Namnabat and M. M. Homayunpoor. (2004). Letter-to-Sound in Persian Language Using Multy Layer Perceptron Neural Network. *Iranian Electrical,*iran, NO. 3 pp:147-154.

Table 1. Expanding Abbreviations (a) and Acronyms (b) in Persian

Abbreviations	Expanded form in Persian
B.B.C	بـی .بـی . سـی
F.B.I	اف . بـی . آی

(a)

Acronyms	Expanded form in Persian
AIDS	ایـدز
CIA	سـیـا

(b)

Table 2. The interaction of stress on Persian terms

Persian term	Pronunciation	meaning
بـشکن	Béškæn	“Break!”
	beškǽn	“snap (of finger)”
بـرو	bóro	“Go!”
	boró	“fast [car, etc.]”
بـردم	bórdæm	“I took.”
	bordǽm	“I’ve taken.”
تمـیز	tæmíze	“It’s clean.”
	tæmizé	“the clean one”
خـوردمـش	xórdæmeš	“I ate it.”
	xordǽmeš	“I’ve eaten it.”

Table 3. First experiment results

	Speech Naturalness	Overall Quality
Word Based TTS System	2.9	2,3
Diphone Based TTS system	4.1	3.12
Triphone Based TTS System	3.2	2.79

Table 4. Second experiment results

		WCR	SCR
Word Database	Sentecces	76	77.3
	Independent words	81	79.3
Diphone Database	Sentecces	83.67	88
	Independent words	89.4	92.3
Triphone Database	Sentecces	86.9	97.8
	Independent words	94.56	98.3

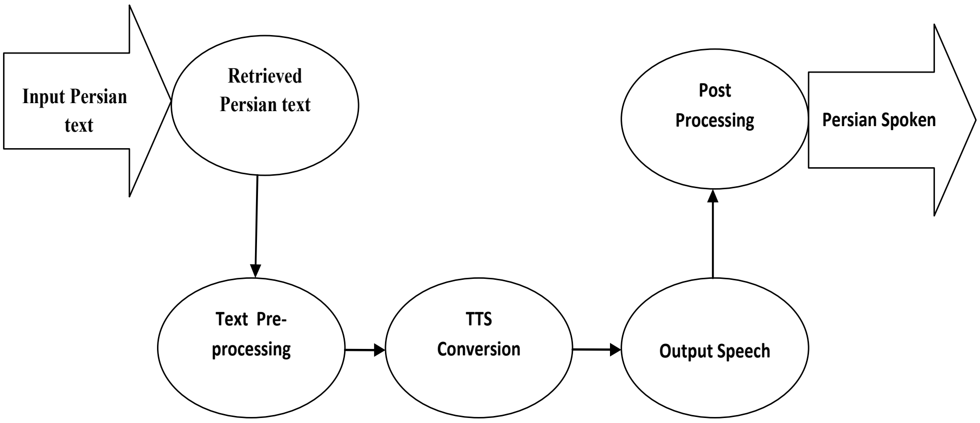


Figure 1. The general architecture of the Persian Text-To-Speech system

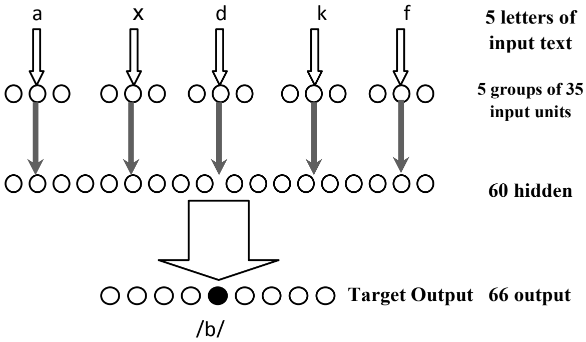


Figure 2. The Neural Network Architecture

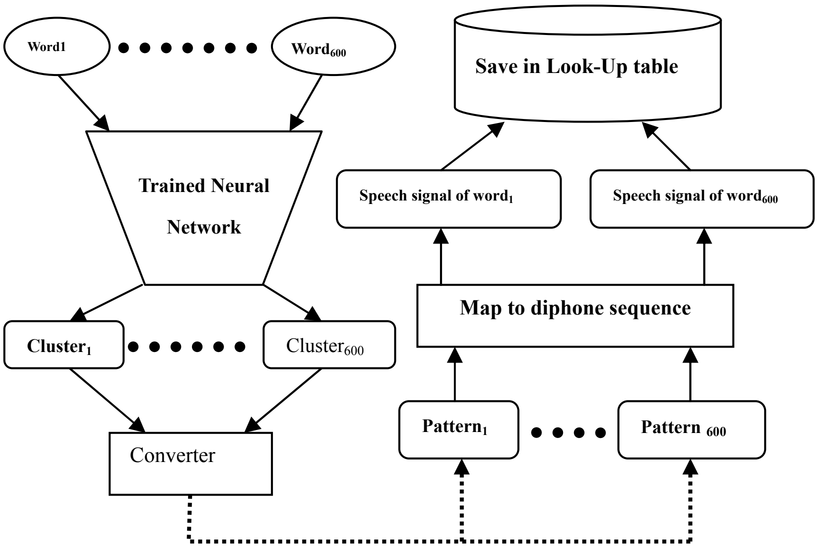


Figure 3. Training the words of the Database

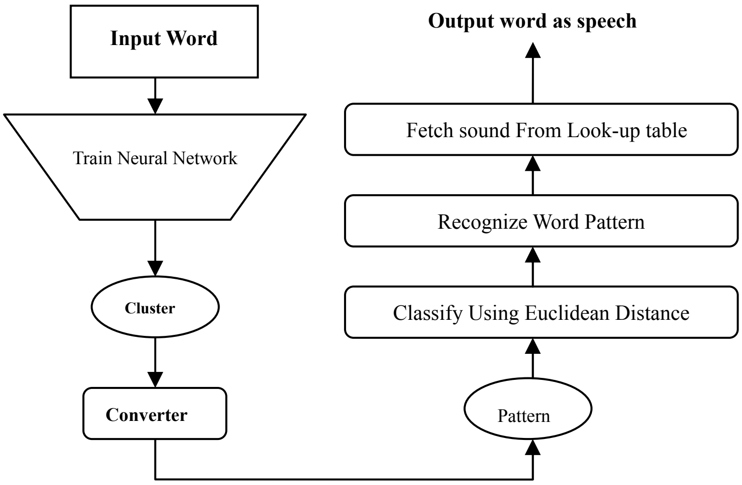


Figure 4. The Word Based TTS System.

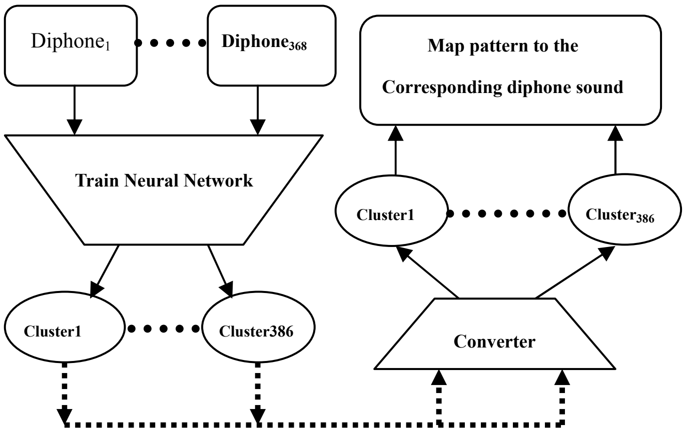


Figure 5. Training the diphones of the Database

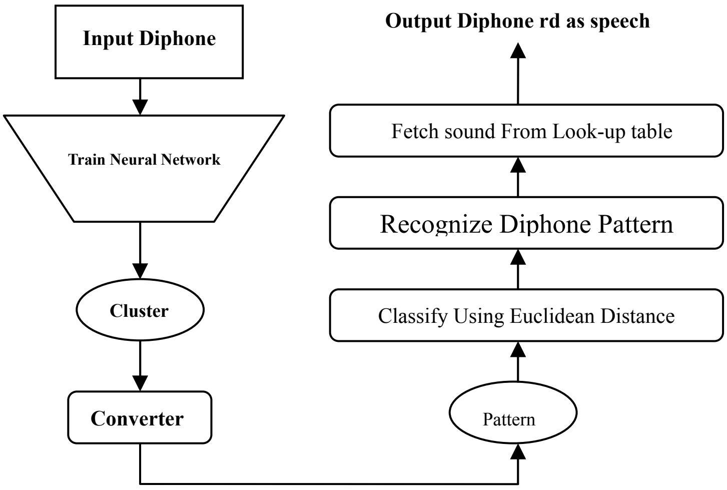

Input Diphone
Train Neural Network
Cluster
Converter
Pattern
Classify Using Euclidean Distance
Recognize Diphone Pattern
Fetch sound From Look-up table
Output Diphone rd as speech

Figure 6. The diphone Based TTS System

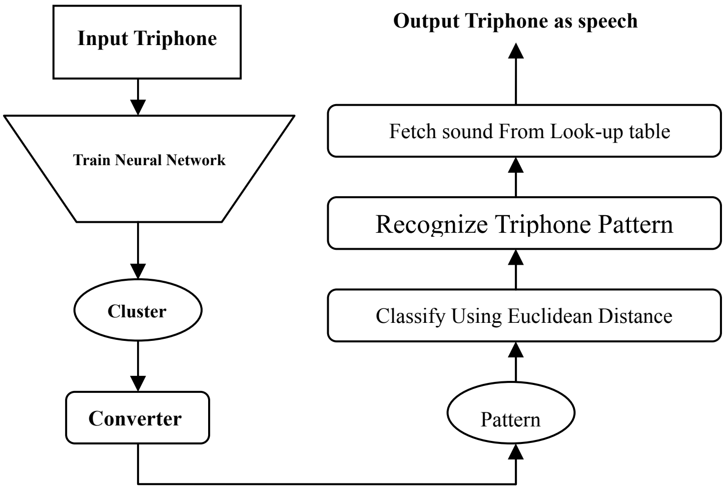
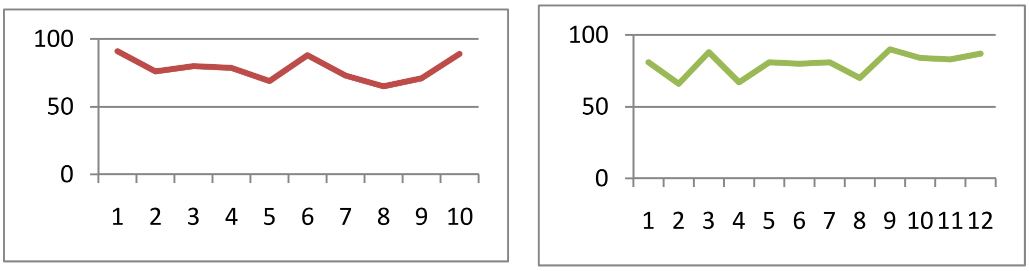
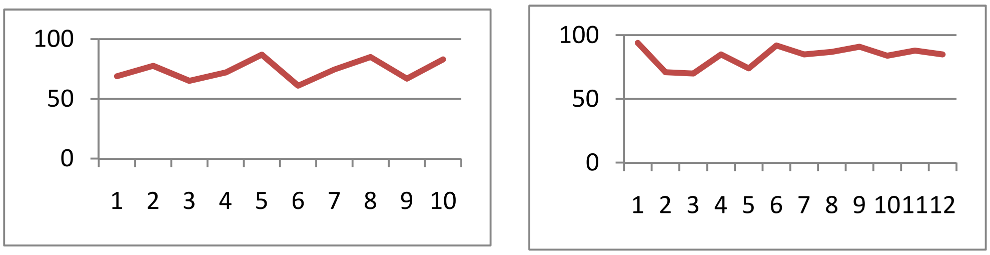

Input Triphone
Train Neural Network
Cluster
Converter
Output Triphone as speech
Fetch sound From Look-up table
Recognize Triphone Pattern
Classify Using Euclidean Distance
Pattern

Figure 7. The triphone Based TTS System



(a) Sentences (b)Discrete words

Figure 8. Diphone recognition accuracy over twelve independent words and ten sentences



(a) Sentences (b)Discrete words

Figure 9. Triphone recognition accuracy over twelve independent words and ten sentences