

# Construction of a Persian Letter-To-Sound Conversion System Based on Classification and Regression Tree

**Mohammad Mehdi Arab**  
Zaban Avaran Pars(ZAP)  
Mashhad, Iran  
mahdi.arab@gmail.com

**Ali Azimizadeh**  
Zaban Avaran Pars(ZAP)  
Mashhad, Iran  
aazimizadeh@gmail.com

## Abstract

Persian writing system, like all other Arabic script-based languages, is special because of omission of some vowels in its standard orthography.

Lack of these vowels causes some problems in Text-To-Speech systems because full transcription of words is needed for synthesis. Then construction of a Letter-To-Sound conversion system is necessary for Text-To-Speech systems because it is not possible to list all words of a language with their corresponding pronunciation in a lexicon.

In this paper, we have presented a Persian Letter-To-Sound conversion system based on Classification and Regression Tree.

The training data is a lexicon of 32,000 words with their corresponding pronunciation which is extracted from Persian linguistic database corpora. The CART is built with Wagon that is a tool of Edinburg Speech Tools for constructing decision trees in Festival.

The final accuracy of this system is 93.61 %, which means that this system is able to predict Persian words' pronunciation comparatively by a high accuracy in comparison with the same system for English which is 94.6% accurate to predict English words' pronunciation in Festival. Also accuracy of the implemented Persian Letter-To-Sound system in festival is more than other previous systems which are implemented out of Festival.

## 1 Introduction

Mapping from strings of letters to strings of sounds is one of the essential parts of Text-To-Speech (TTS) systems. The primary TTS systems used large lexicons for determination of word's pronunciation. However, lexicon of such systems was

large. Also it is not possible to list all words of a language in lexicon then, construction of a Letter-To-Sound (LTS) conversion system is important.

Importance of LTS conversion systems increases for Arabic script-based languages like Persian because of omission of some vowels in their standard orthography.

Generally, there are two major methods for letter-to-sound conversion. The first is based on using some hand written phonological rules. For example in Festival Speech Synthesis system (Black et al., 1999), a basic form of a phonological rule is as follows:

```
(LEFTCONTEXT [ITEM] RIGHTCONTEXT = NEWITEMS)
```

It means that if ITEM appears in the specified right and left context then the output string is to contain NEWITEMS. Any of LEFTCONTEXT, RIGHTCONTEXT or NEWITEMS may be empty. An example is (# [ch] C = k). The special character # denotes a word boundary, and the symbol C denotes the set of all consonants. This rule states that a *ch* at the start of a word followed by a consonant is to be rendered as the k phoneme (Black et al., 1999).

Writing letter to sound rules by hand is hard and time consuming, an alternate method is also available in festival where a Letter-To-Sound system may be built from a lexicon of the language. This technique has successfully been used from English (British and American), French and German (Black et al., 1999). This method is based on computational model of pronunciation, which extracts from training data using a statistical method. The statis-

tical method is Classification and Regression Tree (CART) in Festival.

One of the major previous systems for Persian LTS conversion is based on Statistical Letter to Sound (SLTS) that is implemented by (Georgiou et al., 2004) in University of Southern California. The statistical model, which is used in their project, is Hidden Markov Models (HMM) and the best result of their system is 90.6%.

The other work is done by Namnabat and Homayounpour in Amirkabir University of Technology. They have constructed a system including a rule based section and multi layer perceptron (MLP) neural network and the ultimate accuracy of their system is 87% (Namnabat and Homayounpour, 2006).

We have constructed Persian LTS system as an independent module in Festival by using Wagon, which is part of Edinburg Speech Tools (Taylor et al., 1998). This LTS system is a part of Persian TTS system called ParsGooyan which is implementing in Festival Speech Synthesis system. The system accuracy is 93.61% to predict Persian words' pronunciation.

For Homograph disambiguation and "Ezâfe" clitic determination, there are two independent modules in ParsGooyan TTS system, so disambiguation of homographs' pronunciation and "Ezâfe" clitic determination, is completely out of scope of Persian LTS Conversion module.

Note that in this paper, words or letters, which are bounded with single quotes, are Persian to English letter mapping, and words or letters, which are bounded with double quotes, are Persian words or letters corresponding transcription (Phonetic).

The second part of this paper is devoted to a brief description of Persian orthography and phonology. In the third section, we will address data preparation for training task. In the forth section decision tree method that is used for constructing this system is presented and in fifth section implementation of system is explained. Also in section six, evaluation of the system is presented and finally in section seven conclusion of this study is discussed.

## 2 A Brief Overview of Persian Orthography and Phonology

Persian is an Indo-European language with a writing system like Arabic script. The Persian writing system is a consonantal system with 32 letters in its

alphabet (Windfuhr, 1990). Persian alphabet is listed below.

ا	ب	پ	ت	ث	ج	چ	ح
خ	د	ذ	ر	ز	ژ	س	ش
ص	ض	ط	ظ	ع	غ	ف	ق
ک	گ	ل	م	ن	و	ه	ی

All except four of these letters (including /ا, /پ, /ژ, /گ, /چ, /), borrowed directly from Arabic. In addition, some of these letters were borrowed without their corresponding articulation. As you see below, letters in one row are same in articulation while articulation is different in Arabic.

<u>articulation</u>	<u>Letters</u>
"t"	/ت, /ط
"q"	/ق, /غ
"h"	/ح, /ه
"s"	/ث, /ص, /س
"z"	/ظ, /ض, /ذ, /ز

The sound system of Persian is quiet symmetric. The phonemic system of Persian consists of 29 phonemes composed of 6 vowels (3 long vowels including "i", "u", "â" and three short vowels including "a", "e", "o") and there are 23 consonants and there are also two diphthongs including "ou" and "ei" (Meshkato-dini, 1985). Place of articulation for Persian vowels are listed below but for place of articulation of Persian consonants, please refer to appendix.

<u>Part of Tongue</u>	Front	Back
<u>Tongue Height</u>		
High	(ای) i	(او) u
Mid	(ِ) e	(ُ) o
Low	( ) a	( آ ) â

Persian syllables are always in one of these patterns, CV, CVC, and CVCC. Occurrence of two vowels in one syllable is impossible so number of syllables is almost equal to the number of vowels (Samare, 1986).

In Persian script like other modern scripts of Arabic, diacritics are omitted from writing system. Especially three short vowels are usually hidden in Persian writing system while long vowels are not completely hidden but they don't have their corresponding sound in some contexts. For example, letter /ی/ corresponds to the vowel "i" in words like /ریز/ while here /ی/ is a vowel, or it may sound "y" in a word like /چای/ while /ی/ is a consonant. Table 2 in appendix, illustrates sound variation of some letters.

In Persian orthography, some letters are completely borrowed from Arabic and most of the words that contain these letters are pure Arabic words. These letters are illustrated in table 3.

Finally the last issue that is important to mention about Persian writing system is that, in Persian when two identical letters are placed side by side and the first letter is "sâken" (unvocalized), the first letter is omitted and a gemination sign (tašdid / ˆ / ) will be placed on the second letter. For example /بنتا/ 'bannâ' converts to /بناˆ/ 'banâ+ gemination sign'. The effect of "tašdid" on pronunciation of the phone is that duration of this phone will be approximately doubled with "tašdid" (Samare, 1986). However, in most of Persian standard texts including books, magazines and newspapers, "tašdid" is omitted except for disambiguation.

### 3 Training Data

In order to train LTS systems, a textual database consist of letters with their corresponding pronunciations is required. In the other hand a pronunciation dictionary is required for training task.

#### 3.1 Pronunciation Dictionary

An important issues in providing a pronunciation dictionary for LTS training task, is selecting different words from various contexts. A worth work in Persian corpus Development is Persian Linguistic Database (PLDB) which is done in Institute for Humanities and Cultural Studies. The database is composed of various corpora including newspapers, stories, medical, philosophy, historical, etc.

For providing pronunciation dictionary, first, PLDB corpora normalized.

#### 3.2 Text Normalization

Persian writing system allows certain morphemes to appear either as bound to the host or as free affixes – free affixes could be separated by a final form character (the control character \u200C in Unicode, also known as the zero-width non-joiner) or with an intervening space. The three possible cases are illustrated below for the plural suffix /ها/ "-hâ" and the imperfective (durative) prefix /می/ "mi-". As shown, the affixes may be attached to the stem, they may be separated with the final form control marker, or they can be detached and appear with intervening whitespace. All of these surface forms are attested in various Persian corpora (Megerdoo-mian, 2006).

So, free affixes must be attached to their preceding or following words to prevent errors that may occur in LTS conversion system. However, this work is

<u>Attached</u>	<u>Final Form</u>	<u>Intervening Space</u>
کتابها	کتاب‌ها	کتاب ها
میروند	می‌روند	می روند

not simply possible because some affixes are homograph or homonym. For example the word /تر/ "tar" may be either a noun which means *wet* or it may be suffix /تر/ "-tar" which is a comparative adjective marker. Also the word /می/ "mi" can be pronounced "mey" which means *wine* or it may be durative prefix if is pronounced "mi".

For the first step of text normalization, Persian letters are mapped into appropriate English letters. Then tokenizer extracts the words by attention to space and punctuations. In the next step, system reattaches the affixes to their preceding or following words by considering two attaching strategies.

In the first strategy copulas, plural marker, reduced pronouns and other affixes, which are not homograph or homonym, just attach normally to their preceding or following words. But in second strategy, words like /تر/ "tar" and /می/ "mi" which are homograph or homonym, attach by using a pre trained model. This model is implemented using decision tree that designed based on our training data, which was annotated manually for a correct attaching, or detaching of an ambiguous affix in a proper context.

For example in the following context the word /می/ “mi” must be attached to its following word “khoram” ‘qazâ mi khoram → qazâ mi-khoram’:

/غذا می خورم → غذا میخورم (I eat food)

A sample record of the training data for tokenization is shown below which is related to word / می / “mi”:

((Boolean attach\_sign) ("غذا") ("می") ("خورم")).

In addition, several token to word rules were applied to convert non-standard tokens like phone numbers, dates, abbreviations, etc to standard text.

After normalizing PLDB corpora, the most 41,000 frequently words extracted automatically by SPSS software. About 9,000 of these 41,000 words were omitted because, they were not appropriate for training task and 32,000 pure words remained. Some omission cases are as follows:

1. Pure Arabic words like /الرحمن/ which are seen in some Persian text as religious words.
2. Some tokens that were results of bad typing like /شدهاست/.
3. Some Homographs like /سبک/ that can be pronounced in two forms: “sabk” means style or “sabok” means *light*.
4. Some non-Persian nouns like /بازل/ “bâzel” (*name of a city in Switzerland*).
5. Words which had less than 3 letters such as /ان/, /از/, /در/.

For the last step of providing pronunciation dictionary, we selected appropriate ASCII characters for letter mapping and transcribing, then a native speaker of Persian who was expert in NLP, transcribed mentioned 32,000 words manually, and a lexicon created. The format of the lexicon is one word with its corresponding pronunciation per line as you see below. ‘Nil’ is an empty field for additional feature of word like Part-Of-Speech tag that is not used in our LTS conversion system.

("rft" nil (r a f t))

("rftar" nil (r a f t a^ r))

("rftarC" nil (r a f t a^ r a s^))

("rftarCan" nil (r a f t a^ r e s^ a^ n))

("rftarha" nil (r a f t a^ r h a^))

("rftarhay" nil (r a f t a^ r h a^ y e))

("rftarhayC" nil (r a f t a^ r h a^ y a s^))

As you see in examples, this lexicon contains most of derivational and inflectional forms of a stem. For example in lexicon, we have /رفتار/, /رفتارش/, /رفتارشان/, /رفتارهای/, /رفتارها/, /رفتارهایش/ and so on, that all of them derive from /رفت/.

## 4 Decision Trees

A decision tree is a tree whose internal nodes are tests (on input patterns) and whose leaf nodes are categories (or patterns) (Nilsson, 1996). An example of a decision tree is shown in figure 1. Several systems for learning decision trees have been proposed that CART (Brieman et al., 1984) is one of them.

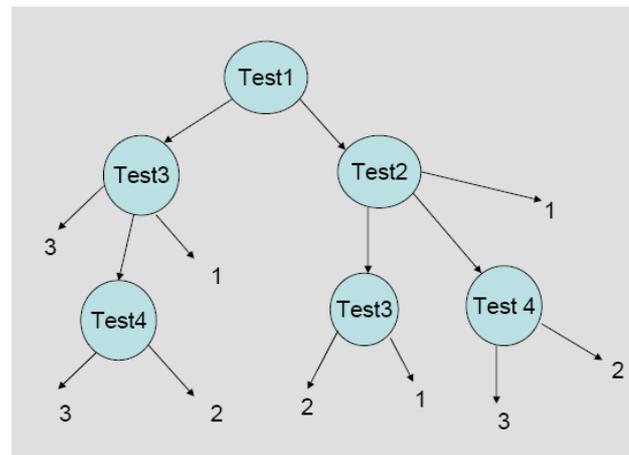


Figure 1 the Example of Decision Tree

### 4.1 CART

One of the basic tools available With Festival Speech Synthesis System, is a system for building and using Classification and Regression Trees (CART) based on (Brieman et al., 1984).

This statistical method can be used to predict both categorical and continuous data from a set of features. The tree contains yes/no questions about features and ultimately provides either a probability distribution, when predicting categorical values (classification tree), or a mean and standard deviation when predicting continuous values (regression tree).

A graphical representation of the statistical method named CART is illustrated in figure 2. In this figure

$\alpha$  may be a probability distribution or a mean and standard deviation.

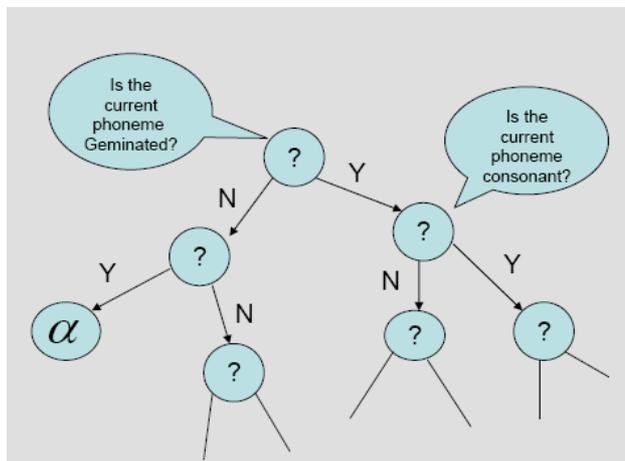


Figure 2 an Example of A CART

## 5 Implementation

### 5.1 Wagon

The program, developed in conjunction with the Festival, called *wagon*, distributed with the Speech Tools, provides a basic but ever increasingly powerful method for constructing trees (Black et al., 1999).

There are many parameters for building decision trees (or decision lists) with *Wagon*. Here we just focus on one of them which are called *Stop*.

In the most basic forms of the tree-building algorithm, a fully exhaustive classification of all samples would be achieved. This, of course is unlikely to be good when given samples that are not contained within the training data. Thus, the object is to build a classification/regression tree that will be most suitable for new unseen samples. The most basic method to achieve this is not to build a full tree but require that there are at least  $n$  samples in a partition before a question split is considered. We refer to that as the *Stop* value (Taylor et al., 1998). Note that number of *Stop* depends on amount of data.

Also *Wagon* requires data and a description of it. A data file consists of set of samples, one per line each consisting same set of features. By default, the first feature is the predictee and the others are used as predictor. The data description consists of a pa-

renthesized list of feature descriptions. Each feature description consists of the feature names and its type (and/or possible values).

### 5.2 Training Process

Building letter to sound CART involves following steps in Festival (Black and Lenzo, 1999):

1. Pre-processing lexicon into suitable training set.
2. Defining the set of allowable pairing of letters to phones.
3. Constructing the probability of each letter/phone pair.
4. Aligning letters to an equal set of phones.
5. Extracting the data by letter suitable for training.
6. Building CART models for predicting phone from letters.

All except the first two stages of this are fully automatic.

The first stage is preprocessing lexicon. Preparation of lexicon explained in part 2.1. 90% of lexicon is used as training data and 10% left as test data.

In the second step, we defined set of allowable. Part of Persian allowable set is shown below.

```
(b_epsilon_b b-a b-e b-o b-i)
(p_epsilon_p p-a p-e p-o p-i)
(t_epsilon_t t-a t-e t-o t-i)
(E_epsilon_s s-a s-e s-o s-i)
(x_epsilon_x x-a x-e x-o x-i)
(e_epsilon_a e o u y e-i )
(q_epsilon_q q-a q-e q-o q-i)
(f_epsilon_f f-a f-e f-o f-i)
(k_epsilon_k k-a k-e k-o k-i)
(n_epsilon_n m n-a^ n-a n-e n-o)
```

Allowable set shows legal conversion of letter to phones. for example a letter like 'v' ( و ) may converts to these phones: 'v-a', 'v-e', 'v-o', 'o', 'u', *epsilon* and "v" itself depending of its context. *Epsilon* means that letter may go to the no phone. For example in a word like 'xvd' /خود/, 'x'

will goes to ‘x-o’ then ‘v’ goes to *epsilon* and ‘d’ will goes to ‘d’.

It is worth to mention that pronunciation restrictions are contemplated in defining allowable set. For example as you see above, for letter ‘n’ / ن/, conversion from phone “n” to “m” is valid. This is because of some pronunciation restrictions. For example, native speakers of Persian does not pronounce “n” in a word like /پنبه/ “panbe” but they pronounce “m” instead “n” then the correct pronunciation is “pambe” rather than “panb”.

After defining allowable set, lexicon must be cumulated. This counts the number of times each letter/phone pair occurs in allowable alignment.

The third stage is constructing probabilities of each letter/phone pair. A part of these probabilities that pertains to letter ‘p’ is shown below:

(p  
 (\_epsilon\_ 0)  
 (p. 0.390799)  
 (p-a. 0.274818)  
 (p-e. 0.113075)  
 (p-o. 0.115254)  
 (p-i. 0.106053))

These probabilities are interesting to review because they contain some basic information. As you see above, conversion of letter ‘p’ to phone “p” is the most probable than other conversions by attention to its probability value (0.390799).

In forth stage each word aligned to an equally lengthed string of phones and in the fifth stage, suitable features were extracted for wagon to build models. Finally, in the last stage we used wagon to build the CART models.

## 6 Evaluation

Training is done by Wagon with three different *Stop* values including 1, 3, and 10 and Test is done automatically by *wagon\_test*, one of Speech Tools programs. The testing results are listed in following table.

STOP VALUE	1	3	10	AVERAGE
ACCURACY	93.61 %	91.46 %	89.54 %	91.54 %

As you see in table, the average accuracy of this system is 91.54 % but we selected the first model

because it is the most accurate model. It is clear that 93.61 % means this system is able to predict pronunciation of about 93 unseen words from 100 words correctly, that seems acceptable for Persian.

## 7 Conclusion

In this paper, we presented a Persian LTS conversion system that is implemented in Festival and is 93.61% accurate which seems very nice attention to this subject that the same model in festival for English is 94.6 % (Black and Lenzo, 1999) accurate to predict pronunciation of unknown English words.

In comparison with two other similar works in Persian LTS conversion systems (Georgiou et al., 2004), (Namnabat and Homayounpour, 2006), this work seems better than others because of following reasons.

The training data of this work is extracted from various corpora which cover better distribution of phones, but two other works relied on public available sources such as Hamshahri newspaper (Georgiou et al., 2004), (Namnabat and Homayounpour, 2006).

Accuracy of this system is more than previous systems which are implemented out of Festival.

The other factor is aspect of implementation of this system in Festival, which implies on being a practical system in developing a Persian speech synthesizer. Also this system is flexible in order to interact with other modules like morphological analyzer (Azimizadeh and Arab, 2007), and part-of-speech tagger modules (Azimizadeh and Arab, 2008), homograph disambiguation module and “Ezâfe” clitic determination module using Festival utterance structure (Black et al., 1999) in order to predict pronunciation of words in textual context rather than isolated words.

Accuracy of this system will probably increase by increasing training data.

## References

- Georgiou P.G., Shirani Mehr H. And Narayanan S.S. (2004). *Context Dependent Statistical Augmentation of Persian Transcripts*. In Proceedings of ICSLP, Jeju, Korea.
- Namnabat M. and Homayounpour M.M. (2006). *A Letter to Sound System for Farsi Language Using Neural Networks*, in proceeding of ICSP.
- Taylor P., Caley R. and Black A.W. (1998). *The Edinburgh Speech Tools Library*. The Centre for Speech Technology Research, University of Edinburgh, 1.0.1 edition. [www: http://www.cstr.ed.ac.uk/projects/speechtools.html](http://www.cstr.ed.ac.uk/projects/speechtools.html)
- Black, A.W., Taylor P.A. and Caley R., (1999). *The Festival speech synthesis system 1.4.0*. [www: http://www.cstr.ed.ac.uk/projects/festival/manual/festival-1.4.0.ps.gz](http://www.cstr.ed.ac.uk/projects/festival/manual/festival-1.4.0.ps.gz)
- Windfuhr G. (1990). *Persian, In Comrie, Bernard (ed.), the world's major Languages*, New York: Oxford.
- Meshkatod-dini M. (1985). *Sound Pattern of Language (An Introduction to Generative Phonology)*. Ferdowsi University Press, Pages 10-112.
- Samareh Y. (1986). *Phonology of Farsi Language*, Markaze Nashre Daneshgahi publisher, Tehran, Pages 35-169.
- Jurafsky D. and Martin J.H. (1999). *An Introduction to Natural Language Processing, Computational Linguistic and Speech Recognition*, prentice hall publisher, pages 91-109.
- Persian Linguistic Database (PLDB), [www: http://pldb.ihs.ac.ir/](http://pldb.ihs.ac.ir/)
- Megerdoomian K. (2006). *Extending Persian Morphological Analyzer to blogs*. In proceeding of Persian language and computer, Tehran University, Iran.
- Nilsson N.J. (1996). *Introduction to Machine Learning*. [www: http://robotics.stanford.edu/people/nilsson/mlbook.html](http://robotics.stanford.edu/people/nilsson/mlbook.html)
- Breiman L., Friedman J., Stone C.J. and Olshen R.A. (1984). *Classification and Regression Trees*. Chapman & Hall/CRC.
- Black A. and Lenzo K. (1999). *Building Voices in the Festival Speech Synthesis System*, unpublished document, Carnegie Mellon University. [www: http://www.cstr.ed.ac.uk/projects/festival/docs/festvox/](http://www.cstr.ed.ac.uk/projects/festival/docs/festvox/)
- Azimizadeh A. and Arab M.M. (2007). *The Persian Morphological Parser with Using POS Tagging*. In proceeding of CAASL2, Stanford University.
- Azimizadeh A., Arab M.M., (2008). *Persian Part Of Speech Tagger Based on Hidden Markov Model*, 9th International Conference on the Statistical Analysis of Textual Data, France.

## Appendix

	bilabial	Labio-dental	Alveolar	Palatal	velar	uvular	glottal
<b>Voiceless</b>	پ [p]		ت، ط [t]		ک [k]		ء [ʔ]، ع [ʕ]
<b>voiced</b>	ب [b]		د [d]		گ [g]	ق [q]، غ [ɣ]	
<b>Fricative-voiceless</b>		ف [f]	س، ص [s]	ش [ʃ]	خ [x]		ح [h]، ه [h]
<b>Fricative-voiced</b>		و [v]	ز، ض، ذ، ظ [z]	ژ [ʒ]			
<b>Affricative-voiceless</b>				چ [tʃ]			
<b>voiced</b>				ج [dʒ]			
<b>lateral</b>			ل [l]				
<b>Flap</b>			ر [r]				
<b>nasal</b>	م [m]		ن [n]				
<b>Glide</b>				ی [j]			

Table 1 Place of articulation for Persian consonants

Letters	Sounds	example
'a' / ا /	a	"abr" (ابر) (cloud)
	e	"esm" / اسم / (name)
	o	"omid" / امید / (hope)
	â	"divâr" / دیوار / (wall)
'v' / و /	o	"xod" / خود / (self)
	u	"ruz" / روز / (day)
	v	"vazir" / وزیر / (queen)
	no sound	"xâhar" / خواهر / (sister)
'y' / ی /	i	"riz" / ریز / (tiny)
	y	"čây" / چای / (tea)
	â	"isâ" / عیسی / (christ)
'h' / ه /	e	"madrese" / مدرسه / (school)
	h	"mâh" / ماه / (moon)

Table 2 Sound Alternation of Some Letters

Borrowed Letters	Sounds	Example
ا	an	"sâniyan" / ثانیاً / (second)
ا	? or No sound	"mosta?jer" / مستأجر / (tenant)
د	a or ? or No sound	"masale" / مسئله / (problem)
ء	No Sound	"enšâ" / انشاء / (composition)
و	a	"moaser" / مؤثر / (valid)

Table 3 Non-Persian Letters