International Journal of Information Science and Management
Vol. 21, No. 1, 2023, 305-317
DOI: 10.22034/ijism.2022.1977618.0 / https://dorl.net/dor/20.1001.1.20088302.2023.21.1.18.8

Original Research

Automation of the Spoken Poetry Rhyming Game in Persian

Mahmood Bijankhan
Faculty of the Literature and Humanities, Laboratory of Linguistics, University of Tehran, Tehran, Iran.
mbjkhan@ut.ac.ir
ORCID iD: https://orcid.org/0000-0002-4175-6854

Hadi Veisi
Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran.
Corresponding Author: h.veisi@ut.ac.ir
ORCID iD: https://orcid.org/0000-0003-2372-7969

Received: 03 December 2021
Accepted: 09 May 2022

Abstract

This paper aims to investigate how a Persian spoken poetry game, called Mosha'ere, can be computerized by using a Persian automatic speech recognition system trained with read speech. To do this, the text and recitation speech of the poetries of the great poets, Hafez and Sa'di, were gathered. A spoken poetry rhyming game called Chakame, was developed. It utilizes a context-dependent tri-phone HMM acoustic modeling trained by Persian read speech with normal speed to recognize beyts, i.e., lines of verses, spoken by a human user. Chakame was evaluated against two kinds of recitation speech: 100 beyts recited formally at the normal rate and another 100 beyts recited emotionally hyperarticulated at a slow rate. About 23% difference in WER shows the impact of the intrinsic features of emotional recitation speech of verses on recognition rate. However, an overall beyt recognition rate of 98.5% was obtained for Chekame.

Keywords: Rhyming Game, Mosha'ere, Persian poetry, Persian Automatic Speech Recognition.

Introduction

Mosha'ere refers to a Persian Poetry rhyming game in which at least two human players participate. According to Milani (2008), one player recites a line of a poem, and a second player must then recite from memory another poem in which the first letter of its first word is the same as the last letter of the last word of the line recited by the previous player. Whoever comes up with a fitting line within a limited time is eliminated. Milani's description confines Mosha'ere to a human-to-human analogue literary game.

Each line of a poetry, called 'beyt' in Persian, consists of two half lines, each of which obeys almost same pattern of short and long syllables. Beyts of poetry rhyme when they end with the same syllable with almost identical vowels and consonants.

This paper reports the implementation of an automatic speech-to-speech Mosha'ere system combining speech signals and poetry text to create a literary game belonging to the domain of digital poetry communication. In such a system, the recitation of a beyt is input to a computer by one human player. As a second player, the computer converts the speech signal of the recited beyt to the corresponding text. Then it outputs a recitation of another beyt of poetry from memory, within a limited time, in which the first letter of its first word is the same as the last letter of the last word of the beyt recited by the human player. Mosha'ere continues until the computer or player loses the game, i.e., one fails to find a suitable beyt to recite within a defined period. Mosha'ere typology could be based on the computer's speed or human response.

Ensslin (2014) theorizes a typology for digital literary games according to their degree of lucidity and literariness. While lucidity describes mechanics like rules, challenges, risks, to actions, and rewards that form the system of a game, literariness describes works in which spoken or written language plays an aesthetic role, like poetry, fiction, or drama. From Ensslin's ludoliterary point of view, Mosha'ere is a poetry game that combines poetic knowledge of players with a kind of computer game design to determine the degree of mastery of the players in memorizing poetries, to test how fast the players can read a beyt of a poem from working memory, to promote ethical and religious teachings existent in the poetries and to reward the winner of the game. Players (human and machine) concentrate on a single literary/linguistic object, i.e., the last and first letters of a poem, and pay attention to the ludic part of the game, i.e., rewarding. Oral poetic discourse between humans and machines also creates fun and excitement in the human player. Ensslin's literariness of the Mosha'ere system should be found in the essence of spoken poetry, mainly coming from the phonological structure of the poems. Error in automatic speech recognition of the human player poem disrupts the game.

Literary gaming is developed in other languages (Ensslin, 2014), and mosha'ere, a poetry game in Persian, has been developed as computer programs on a text-form basis for personal computers as well as cell phone applications, however, they are not attractive to users due to the difficulties in the interaction. The computer-based mosha'ere will be the more exciting game when user can interact using his/her spoken language. To the best of our knowledge, this research is the first try to develop a speech-form basis of the Persian mosha'ere. The research question focuses on how to implement the live performance of spoken mosha'ere between human and machine in the computing environment, given the poetries of the great poets Hafez and Sa'di.

Section 2 discusses collecting Hafez and Sa'di’s written and spoken poetry data in the continuant. Section 3 describes the fundamental elements of spoken poetry derived from spoken poems' phonological analysis in two domains of analysis: suprasegmental and segmental. Section 4 describes the architecture of the Mosha'ere system and the dictionary structure of the Mosha'ere system with the phonological rules involved in making multiple pronunciations of the entries. Section 5 reports the evaluation of Chakame against two kinds of recitation speech types in terms of speaking style and rate. Section 6 discusses and concludes the findings of the paper.

Poetry Data Gathering

The electronic texts of Hafez and Sa'di's poetries were collected from the Internet Ganjoor Website[1]. Hafez's text contains 495 sonnets or 4,192 beyts, corresponding to 275 Kilobytes in size. Sa'di's poems include two poetries Bustan and Golestan. Bustan contains 4,110 beyts, corresponding to 222 Kilobytes in size. Since Golestan consists of both poems and prose, the poems were manually separated from Golestan text and saved in a separate file. The poetic part of Golestan contains 1,035 beyts, corresponding to 59 Kilobytes in size. Mohammad Ali Forughi's editions of both poetries were used as a yardstick to manually correct and modify spelling errors in electronic texts. No attempt was made to segment the texts into linguistic words that are usually the entries of the Persian lexicon because the recitation of the classical Persian poems could be based on the morphemes, tokens in a text. The texts were automatically normalized in two respects: in the first place, the two Arabic letters KAF (U+0643) and YEH (U+064A) were unified with the corresponding Persian letters KEH (U+06A9) and YEH (U+06CC) to achieve the text encoding unification, and in the second place orthographic variations of each word were also unified to get rid of spelling inconsistency challenge. At the very end of data gathering, both Hafez and Sa'di's texts were put together and became a whole text with 9,448 beyts. Non-standard additional characters created by copying texts from the web and punctuation marks were also removed to prepare the texts for building the lexicon and calculating the language model.

The poetries were recited by the voice actor, Amir Noori, and down-sampled by the rate of 16 KHz with a resolution of 16 bits. The volume of WAV sound files of Hafez's recitation is 16:35, and of the MP3 sound files of Bustan and Golestan poems, recitations are 11:25 and 2:36, respectively. All WAV format files were converted into MP3 format to reduce the volume size of the application software,. In cases of difference between the text and speech of a beyt, the text form was modified based on the speech of that beyt, for the software to be able to play the audio file corresponding to the selected beyt by computer, the text file of each beyt co-indexed with the corresponding audio file. For this purpose, an expert in the Persian literature matched the audio file of each beyt with the corresponding text file by listening to the audio files of all beyts.

Phonological Structure of Classical Poetry

In this section, we describe the phonological structure of classical Persian poetry on which automatic recognition of the verses recited by players depends highly.

3.1 Suprasegmental Structure

The rhythmic structure of each beyt of classical Persian poetry is formed according to an organized and complex system of syllable weight. Hayes (1989) distinguishes four degrees of weight for Persian syllables: a light syllable that ends in a short vowel (CV), a heavy syllable that ends in a consonant or has a long vowel or diphthong (CVC or CVV), a superheavy syllable that ends in a consonant and has a long vowel or diphthong (CVVC) or ends in two consonants and has a long vowel or diphthong (CVCC), and ultraheavy syllable that ends in two consonants and has a long vowel or diphthong (CVVCC). However, the quantitative meters of classical Persian poetries reduce four to just two degrees: a short syllable equal to light syllable and a long syllable consisting of a heavy, superheavy or ultraheavy syllable. Persian meter thus arranges syllables of a beyt according to different patterns generated by the number of syllables. Each rhythmic pattern, called foot *(rokn*, in Persian), is a specific arrangement of short and long syllables. Each half-line concatenates several feet, and two half-lines within a beyt may obey the same meter. Each foot consists of three up to eight syllables (Hayes, 1979, Utas, 2008).

Syllabification and stress have a significant role in word recognition within continuous speech (McQueen, Cutler, Briscoe & Norris, 1995, Spitzer, Liss & Mattys, 2007). Given that each Persian syllable must start with a consonant and Persian words may start with a vowel, then a case of two consecutive words may occur in which the first word ends to a consonant and the second word starts with a vowel. In such a case, the syllabification of an utterance consisting of the two words crosses the word boundaries because the final consonant of the first word moves forward to fill in the empty consonantal position at the beginning of the second word. Therefore, the final syllable of the first word is not aligned with the word's end, and the second word's initial syllable is not aligned with the word's beginning. An example is given in (1) from Hafez (c.1315-1390) (Shariari 1999)[2].

در آسمان نه عجب گر به گفته حافظ   سرود زهره به رقص آورد مـسیحا را   (1)
ɹaɢs ɑvaɹad masihɑ ɹɑ be zohre sorude   daɹ ɑsemɑn na ʔaʤab ɟaɹ be gofteje hɑfez
No wonder if in the heavens, as claims Hafez; Venus' song brings Christ to dancing sprees

Persian content words receive stress on the last syllable and the grammatical words are often stressless. Given that a reciter stresses words within each *rokn* according to the syntactic phrases in a half line of verse, stress detection can help to restrict the word boundaries in verse. However, Persian stress is independent from the syllable quantity. This means that the short syllable can be stressed in a word consisting of short and long syllables. Thus, stress detection does not help to recognize the rhythmic structure of verses, which is irrelevant to this research.

3.2 Segmental Structure

As in daily Persian conversation, recitation involves 23 consonants, six vowels and one diphthong ([ow]). The set of consonants is: {p^{h}, b, t^{h}, d, c^{h}, ɟ, ɢ, ʔ, ʧ^{h}, ʤ, f, v, s, z, ʃ, ʒ, χ, h, m, n, l, ɹ, j} and the set of vowels is: {i, e, a, u, o, ɑ} (Bijankhan, 2018). Vowels and coda consonants are major determinants of the foot composition. Vowels are divided into short vowels, i. e. /e a o/ and long vowels /i u ɑ/.

Consonantal clusters at the end of the syllables violate the sonority hierarchy constraint. Thus they do not help detect word boundaries (while in English). Vowel and consonant rhyming is one of the strongest segmental structures in the Persian meter. By knowing it, beyts could be memorable by reciters, which is a key component of Mosha'ere. However, it is not compulsory that beyts or half lines of a beyt rhyme. To be faithful to the Persian meter, reciters can delete, insert, or change the short or long vowels in a beyt. Such a reciter's choice could impact multiple pronunciations. The segmental structure causes, therefore, challenges for word boundary detection in continuous speech. For example, line (2) is a beyt from Sa'di (c.1208-1291) with two feet: CV.CVV.CVV.CVV. and CV.CVV.CVV.

اگر خود هفت سبع از بر بخوانی   چو آشفتی الف ب ت ندانی
aɟaɹ χod haft sabʔ ʔaz bar beχɑni   ʧo ʔɑʃofti ʔalef be te nadɑni   (2)   ʔ
If you recite Seven Sevenths yourself from memory   when disturbed, you would not know Alef Be The

The reciter could change the short vowel /e/ of the words /be/ and /te/ into the corresponding long vowel /i/ to make the second half line rhythmic pattern the same as the first.

4. Mosha'ere System

In this section, the implementation details of a Mosha'ere system, called *Chakame*, is presented. The system is a two-participant Persian poetry rhyming game played by a human player, called *user* and a computer. The game is started randomly by the user or computer. The flowchart of the system, assuming that the computer is the starter of the game is demonstrated in Figure 1. The computer selects a *beyt* randomly, displays it in text, and plays its corresponding speech signal. The selected beyts by computer in the whole process of Mosha'ere are added to a list called *PlayedList* to avoid computer reselection of beyts.

Afterward, the user can continue the game by reading a beyt, according to the Mosha'ere rule above. The spoken beyt of the user is then sent to customized automatic speech recognition (ASR) module to be converted into its corresponding text. The ASR system used in this project is based on Nevisa Persian speech recognition engine (Sameti, Veisi, Bahrani, Babaali & Hosseinzadeh, 2011), a speaker-independent, very large vocabulary system that utilizes the hidden Markov model (HMM) for acoustic modeling and N-gram for language modeling. The details of the designed ASR system for this project are given in the next section. As the recognizer is tri-phone based and its output does not match the beyts, we have to post-process the recognized word sequence to select the most similar beyt. To this end, a modified Jaccard coefficient is used to calculate the similarity between the output of the ASR and all beyts. The basic Jaccard coefficient to measure similarity between two finite sample sets is calculated as in equation (2) which is defined as the size of the intersection divided by the size of the union of the sample sets.

$$J(O, B_i) = \frac{|O \cap B_i|}{|O \cup B_i|} = \frac{|O \cap B_i|}{|O| + |B_i| - |O \cap B_i|} \tag{2}$$

In this equation, O denotes the set of ASR output tokens (i.e., words) and B_i Indicates the tokens of the i^{th} beyt. In this research, we compute the above equation for three sets of tokens, unigram, bi-gram, and tri-gram tokens of O and B_i. Then, the Jaccard coefficient for these sets are averaged and the maximum is selected as the result (equation 3).

$$i^* = \underset{i}{\operatorname{argmax}} \left(\frac{J_{uni}(O, B_i) + J_{bi}(O, B_i) + J_{tri}(O, B_i)}{3} \right) \tag{3}$$

The most similar beyt is selected as the user's beyt and is shown to the user to confirm its correctness. The Mosha'ere rule, the equality of the first letter of the user's beyt and the last letter of the computer's beyt, is then checked. If the rule is hold, the game continues by computer selection of another suitable beyt.

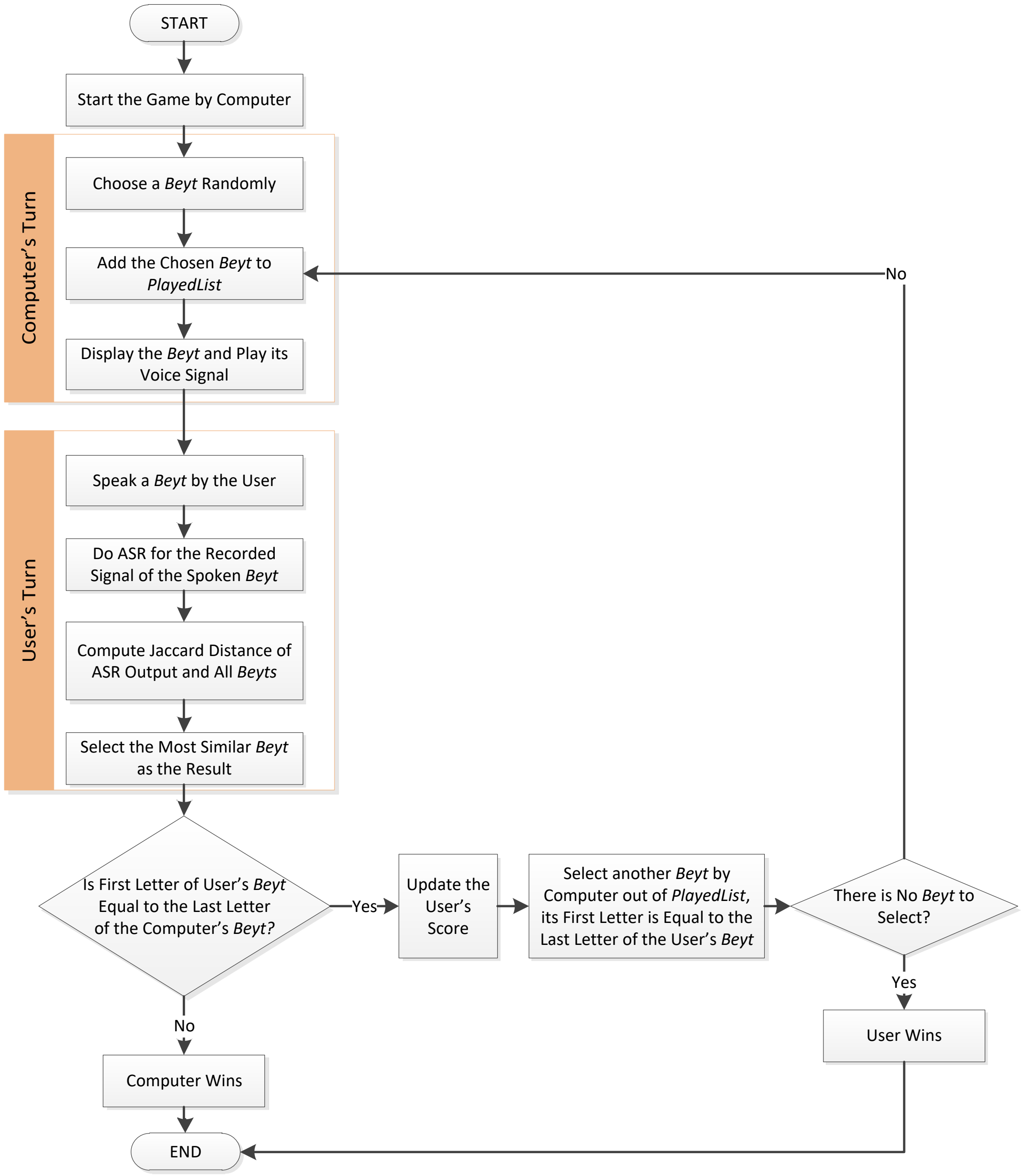
START
Start the Game by Computer
Computer's Turn
Choose a *Beyt* Randomly
Add the Chosen *Beyt* to *PlayedList*
Display the *Beyt* and Play its Voice Signal
User's Turn
Speak a *Beyt* by the User
Do ASR for the Recorded Signal of the Spoken *Beyt*
Compute Jaccard Distance of ASR Output and All *Beyts*
Select the Most Similar *Beyt* as the Result
Is First Letter of User's *Beyt* Equal to the Last Letter of the Computer's *Beyt?*
Yes
Update the User's Score
Select another *Beyt* by Computer out of *PlayedList*, its First Letter is Equal to the Last Letter of the User's *Beyt*
There is No *Beyt* to Select?
No
Yes
User Wins
No
Computer Wins
END

Figure1: The flowchart of the proposed Mosha'ere system (Chakame)

Mosha'ere is a game, and its degree of lucidity should be specified. A scoring method should be defined to gamify the implemented system. Users can utter his/her beyt by a start/stop button in Chakame. The recording is started upon clicking the start button. It is considered that the user answers in 30 seconds. If the system does not receive a signal during this time, the game gets over, and the computer is selected as the winner. Once the user pushes the stop button, the recorded signal is considered the answer and will be sent to the ASR system if its duration is more than 5 seconds. As the speed in answering is a supremacy factor in mosha'ere, the calculation of points for each beyt depends on the user's answer time. We calculate the score of each answer according to equation (4) in which the fast answers receive higher scores. A user's total score during a game is the sum of the scores for each answer.

$$Score = 30 - (Length\ of\ recorded\ signal) \qquad (4)$$

If the uttered beyt from the user is repetitious, a warning message is displayed to the user, and the score of that answer is halved. The current game's score and the user's total scores (for all their games) are demonstrated. According to the total score, the user's level is defined: as beginner, experienced and professional (the poet).

The game gets over, and the user wins if all valid beyts, starting with the last letter of the user's beyt, are played so far. On the other hand, the computer wins if the user's answer does not satisfy the mosha'ere rule.

Automatic Speech Recognition

The ASR module is the main sub-system of the proposed Mosha'ere system, making it possible for the user talks to the computer. A Persian ASR system is utilized to recognize the user's uttered beyt, a customized version of Nevisa ASR engine (Sameti et al., 2011). Nevisa is a commercial speech recognition system that is widely used as a voice dictation software for Persian. There are various types of research for Persian ASR using a well-known HMM-based classic approach (Sameti et al., 2011; Goodarzi & Almasganj, 2016; Hadian, Povey, Sameti & Khudanpur, 2017) or state-of-the-art deep learning technique (Daneshvar & Veisi, 2016; Hajimani, 2017; Ansari and Seyyedsalehi, 2017; Hadian et al., 2017; Hajitabar, 2016; Babaali, 2016). Although experimental results confirm the superiority of the deep learning methods, feed forward and recurrent neural networks, over the HMM-based recognition (Hajimani, 2017; Hadian et al., 2017; Babaali, 2016) and the Nevisa system also has a deep learning-based engine, we have used the HMM-based one in this research. The reason is that the HMM-based engine of Nevisa is a stand-alone application (the deep learning engine is a service) as the Chakame is. It is also easily customizable for the Mosha'ere purpose. Furthermore, the performance of the HMM-based version in Mosha'ere task (i.e., recognizing beyts after the post-processing) is good enough for real applications.

The Nevisa ASR engine is customized for Mosha'ere. The customization mainly includes preparing a lexicon and learning a personalized language model. The acoustic model of Nevisa remains the same for Mosha'ere. The acoustic models of Nevisa are trained using read Persian speech data from Farsdat (Bijankhan, Sheikhzadegan & Roohani, 1994), Large Farsdat (Sheikhzadegan & Bijankhan, 2006) and collected data set by Nevisa developers, a total of more than 100 hours.

Due to the high variations in the pronunciations and the addition of special prosody to the speech of reciters, the creation of a phonetically rich lexicon plays an important role in the recognition. After normalizing the text corpus, the number of total tokens of the poems reached 13,859 entries for building the lexicon. The details of creating a customized lexicon for the system are presented in the next section.

The ASR engine in Chakame samples speech signals in 16 KHz and blocks them in 25ms frames and 15ms of overlaps. The engine's front end is based on Mel-frequency cepstral coefficients (MFCCs) using a pre-emphasis filter of factor 0.97 and Hamming windowing. The spectrum of each frame is weighted and summed up using 40 Mel-scaled triangular filters, and finally, 13 coefficients are calculated after applying logarithm and discrete cosine transform (DCT), respectively. The first and second derivations of the coefficients, delta and delta-delta, are also concatenated to the MFCC features to make a 39 dimensional vector for each frame. Due to the robustness ability of cepstral mean normalization (CMN) method, this technique is applied to the feature vectors for each uttered beyt.

The ASR engine's acoustic models are context-dependent tri-phone-based modeling using HMM. Each HMM is built using five states and eight Gaussian mixtures per state. The HMMs are left-to-right, in which skips and self-loop transitions are allowed. The features are assumed uncorrelated resulting in diagonal covariance matrices. In the tri-phone modeling of 29 Persian phones, the HMM states are tied to four thousand senons. The system utilizes maximum likelihood linear regression (MLLR) and maximum a posteriori (MAP) speaker/environment adaptation methods.

Statistical n-gram methodology was used to model the language of poetries at the word level. In the system, we have used 3-gram modeling and Witten-Bell smoothing. The language model is extracted from a corpus containing all text of poems used in the system. The text corpus of the poems described in Section 2, contains 107,840 tokens.

A summary of the ASR module setting in Chakame system is given in Table 1.

Table 1

Configuration of ASR module of Chakame

Front-End	Sampling Rate	Frame Size	Overlap	Pre-emphasis Factor	# of Mel Filters	# of MFC Coefs	Normalization
	16 KHz	25 ms	15 ms	0.97	40	39	CMN
Modelling	Acoustic Unit	# Tri-Phones	HMM Topology			Lexicon Size	Language Model
	Tri-Phone	4,000	Left-to-Right	Gaussian Distribution	5 States, 8 Mixtures	13,859	3-gram, Witten-Bell smoothing

4.2 The Lexicon

The lexicon is another key component of the Mosha'ere system. It contains 13,859 entries. Each entry could be either a token belonging to a multi-token unit, as in (5), or a multi-unit token, as in (6) (Halteren, 1999).

فرو برد	' he/she swallowed'	(5)
نیک‌بختی	'Good luck'	(6)

The smallest entry is a grammatical morpheme like فرو in (5), and the largest entry is a sequence of a few morphemes. In many cases, to keep recitations consistent with a specified rhythmic pattern of a beyt, each morpheme of a multi-token complex word could be recited with its stress, facilitating automatic recognition of such tokens.

CMU Language Modelling Toolkit[3] was used to extract the lexicon from the whole electronic text. Two items of information are specified for each lexicon entry: Persian written form and the SAMPA transcriptions of multiple pronunciations. The first pronunciation of all entries is the canonical form, formally in isolation. Other pronunciations of each entry were obtained based on the following phonological rules:

1. Ezafe morpheme [e] is added to the end of the entries whose part of speech is noun, adjective or number, provided that the entries do not end with vowel letters Alef ا or Vav و.

2. Some words have more than one phonemic or phonetic form (Hayes 2009), due to alternation between [a] and [e], [j] and [ʔ], or [e, o] and null phoneme.

3. Pronunciation varieties are caused by the harmony of a vowel with another vowel, like

vowel harmony between [o] and [u], [e] and [i], or [a] and [ɑ].

4. Pronunciation varieties caused by the assimilation of a consonant with another one, like assimilation between [n] and [m], [t] and [s], [d] and [z], [ʤ] and [ʃ] or [ɢ] and [χ].

5. Pronunciation varieties caused by non-lexical homographs (Bijankhan, Sheykhzadegan, Bahrani & Ghayoomi, 2011). Such varieties are recognized through the linguistic context of the beyts in which they have been used. For example, رخت means 'clothes' with the pronunciation [ɹaχt], and it may mean 'your face' with the pronunciation [ɹoχat] in the linguistic context of a beyt.

6. Pronunciation varieties caused by articulating the diphthong [ow] in some Arabic loanwords like موعظه [mowʔeze] 'preaching'. Such pronunciations occur in slow, formal and emotive recitations.

7. Pronunciation varieties caused by diphthongs on the boundary between two morphemes within a multi-unit token, one ends with a vowel and the other begins with a vowel, like the pronunciation of the token پیمانه‌ام [pejmɑne+am] 'my bowl' (+ shows the boundary). In such cases, the vowel sequence, i e. hiatus, could be resolved by the glottal stop insertion. Then the pronunciation of پیمانه‌ام would become [pejmɑne+ʔam]. The hiatus could be resolved by deleting one of the vowels. For example, the pronunciation of the token کآب [kɑb] 'that water'.

Performance Evaluation

Since no automated poetry game is found whose methodology can be compared with the methodology we have applied, an experiment is conducted to evaluate Chekame performance.

Recognition of the user's speech plays the most important role in working with Chakame. As described, this is done by ASR module and its related Jaccard similarity post-processing. So, 200 beyts of Hafez and Sa'di's poetries were selected randomly in two parts. The first part contains 100 beyts. Ten Persian speakers, including 5 men and 5 women acquainted with Persian literature and poetry, each recited ten beyts formally at a normal rate. The second part also contains 100 beyts which the voice actor recited emotionally with a slow rate, which makes the automatic speech recognition a rather difficult task because the acoustic model of the ASR module of the system is trained by non-stressed emotion and normal rate speech of large Farsdat database. 200 audio files of the beyts were segmented, each matched with the corresponding text file. These 200 beyts were inputted to the Chakame system and the recognition accuracy rate was measured according to (7), which computes beyt accuracy rate (BAR), i.e., the ratio of the correctly recognized beyts.

$$BAR = \frac{Number\ of\ beyts\ that\ are\ correctly\ recognized}{Number\ of\ total\ beyts} \tag{7}$$

The result of the evaluation is given in Table 2. Although the signal of the first part of the test set was recorded in a real environment (i.e., office) and the second part in a clean environment (i.e., studio), the system's performance on the first part is higher than the second one. This predictable phenomenon is due to the reading style of the second part, which is spoken in a very tonal and slow rate manner.

Table 2

Performance of Chakame in user's speech recognition

Test Set	Speaker	Gender	# of beyts	BAR (%)
First Part	1	Male	10	100
	2	Female	10	100
	3	Female	10	100
	4	Male	10	100
	5	Male	10	100
	6	Male	10	90
	7	Male	10	100
	8	Female	10	100
	9	Female	10	100
	10	Female	10	100
Second Part	11	Male	100	98
Average				98.5%

The performance of the ASR module without using Jaccard similarity post-processing is given in Table 3. This table gives both word accuracy rate (WAR) and beyt accuracy rate. WAR is the ratio of the number of words recognized correctly by the ASR system, which is defined similarly to (7) but for the words. It is, of course, 1-WER (word error rate). Also, the correctness metric is given to show how the system inserts additional words in the output. The WAR and correctness are defined as below, in which N denotes the number of words in the reference beyt, I, D and S define several insertions, deletion, and substitution errors, respectively.

$$WAR = \frac{N-(I+D+S)}{N} \tag{8}$$

$$Correctness = \frac{N-(I+S)}{N} \tag{9}$$

From Table 3, it can be inferred that:

- The accuracy of words in Mosha'ere is relatively lower than the rate reported for reading speech (Sameti et al., 2011). This is due to the mismatch between training data and the evaluation data (i.e., recitation of beyts). The ASR engine is trained with typical read speech at a normal rate and non-stressed emotion speech. At the same time, it is evaluated by recitation speech which is mainly emotionally pronounced with a slow rate and highly dependent on the poetic sense. The word accuracy rate for the first part (80.15%) is remarkably higher than the second part (57.47%) due to the reading style of the poetic sense of recitation of the speaker in the second part.
- The difference between accuracy and correctness shows that the insertion error rate of the system is 6.4%. One-syllable grammatical words were the most frequently inserted words, among others.
- The comparison of the results of this table and Table 2 show that although the word accuracy rate is acceptable, the BAR is not good enough to be used directly in Chakame; thus, the Jaccard post-processing is necessary.

Table 3
Performance of ASR module (without Jaccard post-processing)

Test Set	Speaker	Gender	# of beyts	WAR (%)	Correctness%	BAR (%)
First Part	1	Male	10	84.38	87.59	0.00
	2	Female	10	77.42	80.02	0.00
	3	Female	10	69.56	75.81	0.00
	4	Male	10	80.06	82.94	0.00
	5	Male	10	78.76	82.61	0.00
	6	Male	10	87.10	90.20	0.00
	7	Male	10	85.53	87.30	0.00
	8	Female	10	78.89	84.10	0.00
	9	Female	10	83.71	86.21	0.00
	10	Female	10	76.05	82.11	0.00
Second Part	11	Male	100	57.47	66.58	1.00
Average				68.8	75.2	0.5

Discussion

A context-dependent tri-phone HMM classifier trained by Persian read speech with normal speed was customized to recognize beyts, i.e., verses, spoken by human user in a Mosha'ere system, as shown in Figure 1. The main issue in such a human-machine interactive system is the mismatch between training data and the recitation of beyts. While Mosha'ere ASR system is trained with typical read speech of newspapers with normal rate and non-stressed emotion speech, it was tested against two kinds of recitation speech: 100 beyts recited in an almost formal manner with the normal rate; and another 100 beyts recited in an emotionally hyperarticulated manner with slow rate and highly dependent on the poetic sense that reciter feels. Such mismatch is the most crucial source of ASR performance (Benzeghiba et al. 2007, Meyer, Jürgens, Wesker, Brand & Kollmeier, 2010; Meyer, Brand & Kollmeier, 2011), causing more word error rate, as reported above in Table 3.

Conclusion

Many studies have focused on ASR challenges for spontaneous or conversational speech compared to read speech (Baumann, Kennington, Hough & Schlangen, 2017, among others). Unlike conversational speech, Persian recitation speech data of verses contain no dysfluencies. However, unlike reading speech, they may contain unusual pauses and emotions. Since Persian is a pitch-accent language (Sadat-Tehrani, 2008), content words, phonological phrases, and intonational phrases each have a unique prominent syllable (Kahnemuyipour, 2003). The prominent syllables are predictable in both Persian read and recitation speech, given syntactic and phonological constituents. However, the rhythmic pattern of the long and short syllables of the recited verses impacts the phonemes and the temporal structure because reciters tend to hyperarticulate the long syllables more than the short syllables, regardless of the prominence of the syllables. Each word is pronounced as a whole, with the final syllable stressed in the read speech.

In contrast, it is pronounced as a sequence of short and long syllables in the recitation speech; thus, short syllables could be compressed and reduced much more than long syllables to maintain the foot pattern of the verses. This may lead us to conclude that tri-phone acoustic modeling is more consistent with reading speech than recited verses. A syllable-based ASR could be proposed to model acoustic patterns of recited speech of verses in Persian. Kanda, Lu and Kawai (2016) reported improvement in the WER for a Japanese ASR using 263 Japanese syllables (known as "kana") for the recognition unit. Japanese syllabary is as simple as the Persian syllabary.

We conclude that the intrinsic features of the recited verses in terms of speaking style (non-stressed vs. stressed emotion) or rate (normal vs. slow) impact the WER of an ASR trained with reading speech data. However, the language model of the Mosha'ere system's poetries compensates for the system degradation. The ASR challenge remains if more Persian classic poetries add to the system.

Acknowledgment

The Iran National Science Foundation funded this project under contract number 94023772. However, the responsibility for the content of this study lies with the authors alone.

Endnotes

1. https://ganjoor.net/
2. http://www.hafizonlove.com/divan/01/
3. http://www.speech.cs.cmu.edu/SLM_info.html

References

Ansari, Z. & Seyyedsalehi, S. A. (2017). Toward growing modular deep neural networks for continuous speech recognition. *Neural Computing and Applications*, *28*(1), 1177-1196.

Babaali, B. (2016). Establishing a New and Efficient Platform for Persian Speech Recognition, *Signal and Data Processing Journal*, 13 (3).

Baumann, T., Kennington, C., Hough, J. & Schlangen, D. (2017). Recognising conversational speech: What an incremental asr should do for a dialogue system and how to get there. In *Dialogues with social robots* (pp. 421-432). Springer, Singapore.

Benzeghiba, M., De Mori, R., Deroo, O., Dupont, S., Erbes, T., Jouvet, D., Fissore, L., Laface, P., Mertins, A., Ris, C. & Rose, R.. (2007). Automatic speech recognition and speech variability: A review. *Speech communication*, 49(10-11), 763-786.

Bijankhan M. (2018). Phonology. Chapter 5 of Anousha Sedighi and Pouneh Shabani-Jadidi (2018). *The Oxford Handbook of Persian Linguistics*. Oxford: Oxford University Press.

Bijankhan, M., Sheikhzadegan, J., Roohani, M.R., Samareh, Y., Lucas, C. & Tebyani, M. (1994). FARSDAT-The speech database of Farsi spoken language. In *Proceedings of Australian Conference On Speech Science And Technology*, 2, (pp. 826-831).

Bijankhan, M., Sheykhzadegan, J., Bahrani, M. & Ghayoomi, M. (2011). Lessons from building a Persian written corpus: Peykare. *Language resources and evaluation*, 45(2), 143-164.

Daneshvar, M. & Veisi, H. (2016). Persian phoneme recognition using long short-term memory neural network. In *Eighth IEEE International Conference on Information and Knowledge Technology (IKT)*.

Ensslin, A. (2014). *Literary Gaming*. Cambridge: The MIT Press.

Goodarzi, M. M. & Almasganj, F. (2016). A GMM/HMM model for reconstruction of missing speech spectral components for continuous speech recognition. *International Journal of Speech Technology*, 19(4), 769-777.

Hadian, H., Povey, D., Sameti, H. & Khudanpur, S. (2017). Phone Duration Modeling for

LVCSR Using Neural Networks. In *INTERSPEECH* (pp. 518-522).

Hajimani, A. (2017). *Persian Speech Recognition Using Deep Learning*, M.Sc. Thesis, University of Tehran.

Hajitabar, A. (2016). *Large Vocabulary Isolated Word Recognition Using Deep Neural Networks*. M.Sc. Thesis, Sharif University of Technology.

Halteren, H. V. (1999). *Syntactic Wordclass Tagging* (9). Springer Science & Business Media.

Hayes, B. (1979)**.** The rhythmic structure of Persian verse. *Edebiyat,* 4, 193-242.

Hayes, B. (1989). Compensatory lengthening in moraic phonology. *Linguistic inquiry*, 20(2), 253-306.

Hayes, B. (2009). *Introductory phonology*. Malden: Wiley-Blackwell.

Kahnemuyipour, A. (2003). Syntactic Categories and Persian Stress. *Natural Language and Linguistic Theory,* 21, 333–379.

Kanda, N., Lu, X. & Kawai, H. (2016). Maximum a posteriori Based Decoding for CTC Acoustic Models. In *Interspeech* (pp. 1868-1872).

McQueen, J. M., Cutler, A., Briscoe, T. & Norris, D. (1995). Models of continuous speech recognition and the contents of the vocabulary. *Language and cognitive processes*, 10(3-4), 309-331.

Meyer, B. T., Brand, T. & Kollmeier, B. (2011). Effect of speech-intrinsic variations on human and automatic recognition of spoken phonemes. *The Journal of the Acoustical Society of America*, 129(1), 388-403.

Meyer, B. T., Jürgens, T., Wesker, T., Brand, T. & Kollmeier, B. (2010). Human phoneme recognition depending on speech-intrinsic variability. *The Journal of the Acoustical Society of America*, *128*(5), 3126-3141.

Milani, A. (2008). *Eminent Persians: The Men and Women Who Made Modern Iran.* Syracuse University Press.

Sadat-Tehrani, N. (2008). The Structure of Persian Intonation. In *Proceedings of the Speech Prosody* (pp. 249-252). ISCA Archiv.

Sameti, H., Veisi, H., Bahrani, M., Babaali, B. & Hosseinzadeh, K. (2011). A large vocabulary continuous speech recognition system for Persian language. *EURASIP Journal on Audio, Speech, and Music Processing*, 2011(1), 1-12.

Sheikhzadegan, J. & Bijankhan, M. (2006). Persian speech databases. In *Proceedings of the 2nd Workshop on Persian Language and Computer* (pp. 247–261).

Spitzer, S. M., Liss, J. M. & Mattys, S. L. (2007). Acoustic cues to lexical segmentation: A study of resynthesized speech. *The Journal of the Acoustical Society of America*, 122(6), 3678-3687.

Utas, B. (2008). *Prosody: Meter and Rhyme*. J. T. P. de Bruijn. Publisher: I. B. Tauris & Co Ltd.