Using of Recurrent Neural Network with a Refine Model to Compose Vietnamese Luc-Bat Poems for Human-Robot Verbal Interactions

Nguyen Hai Dang [1] and Nguyen Truong Thinh [2,*]

[1] Department of Mechatronics, Ho Chi Minh City University of Technology and Education, Ho Chi Minh, Vietnam
[2] Institute of Intelligent and Interactive Technology, University of Economics Ho Chi Minh City (UEH), Ho Chi Minh, Vietnam
*Correspondence: thinhnt@ueh.edu.vn (N.T.T.)

***Abstract*—Interactions between humans and service robots require not only the machines' friendly appearance but also the way of communication to be suitable for different cultures. Moreover, service robots that transfer information to their human partner verbally, especially in lyrical ways such as songs or poems help the conversation to be closer, more impressive, and easier to follow. This work presented an implementation containing a recurrent neural network-based model, which was trained directly on semantic verses by collecting 43,031 paragraphs for the dataset to increase the contextual creativity itself to help a service robot compose a Vietnamese traditional six-eight poetry. By also being implemented with a refining algorithm, the program can automatically replace syllables that do not follow the template rules of Vietnamese Luc bat poetry. The making of a poem is a sequential process from word to sentence, and the content of the previously complete sentence will serve as the basis for developing words for the next. Through experiments, it is found that the program performs its task impressively as the uniqueness in arranging the words goes with the diversity in the content of every generated poem.**

***Keywords*—Recurrent Neural Network (RNN), refine model, poem generating, human-robot interaction, verbal interaction, service robot, Vietnamese Luc bat poem, Vietnamese poem generator, robot poetry greetings**

I. Introduction

Robots, especially the service types should have enhanced comprehension ability rather than the "simple commands only" as decades ago, they ought to be more effective, and purposeful in providing any conversations or talks [1]. There have been such applications with both capabilities of verbal and non-verbal interactive communication: Hala, which is a bilingual receptionist robot with cultural sensitivity, shares with listeners about her personal life or weather conditions outside in both English and Arabic [2]. Hala may also nod her head and make facial expressions when talking to the visitors at her desk. Robovie is an autonomous humanoid robot that can interact with people through greetings, body gestures, eye contact, or even some self-behaviors of head-scratching or arms folding. Robovie [3] is equipped with sensors such as vision, sense of touch, and audition as well as with the height of a primary student to make it more interactive with humans. Or Robovie-IV [4], which takes part in the working atmosphere of an office, navigates himself around and allows the others to play with him by touching his nose, patting his head, or even starts chatting with a particular person standing nearby. As mentioned, robots can be closer to human beings through dramatic art, namely Geminoid F performed in a play by reading a poem called "Sayonara" and collaborating with a human actor. Impressively, a number of audiences were surprised by the fact that there had been an android in the play and performed naturally as a human [5]. Summarily, those have shown the importance of effective interaction between robots and humans can be achieved through many approaches, but mainly the verbal one since the natural language is a crucial part of human society. Moreover, robots can now help preserve many traditional to modern types of artwork, especially literature. For this reason, we aimed to make a new turning point for a service robot by enabling it to compose Vietnamese traditional poetry. In this paper, we focus on making a model based on the Recurrent Neural Network (RNN) network to generate Vietnamese Luc bat poetry. It is from this trained model that will then be used for our service robot, which is expected to make the interaction between it and Vietnamese users more friendly, and interesting.

There has been a significant number of works in many genres that have enriched Vietnamese folklore, such as tales, parables, idioms, folk, lyrics and poems. Since the early nineteenth century, authors have focused on the connection between people's daily lives and nature, the relationship between human destiny and their characteristics, and shown many inner aspects of human beings. Namely "Truyen Kieu"—an epic masterpiece of Vietnamese literature written by Nguyen Du (1765–1820) [6], which is created in the form of the six-eight verse telling the whole story of the misfortune life of the main character named Kieu [6]. The acute use of words as

Manuscript received March 30, 2023; revised April 11, 2023; accepted July 20, 2023; published October 27, 2023.

well as the novel comparing imagery have pleased generations of Vietnamese reviewers. The following extract is from a high-ranked translated version (Ross [7]) of version Huynh-Sanh-Thong [6]:

Two daughters, beauties both, had come before
(Đầu lòng hai ả tố nga)
Thuy Kieu was oldest, younger was Thuy Van
(Thúy Kiều là chị, em là Thúy Vân)
Bodies like slim plum branches, snow-pure souls:
(Mai cốt cách, tuyết tinh thần)
Each her own self, each perfect in her way.
(Mỗi người một vẻ, mười phân vẹn mười)

Poetry is one of the valuable spiritual cultural legacies of Vietnam since it documents the nation's evolution from ancient dynasties to the modern days [8]. The traditional poetic structures of Vietnamese literature include the six-eight (the Luc bat), 5-syllable, 7-syllable, 8-syllable, and free-style poetry that make the diversity in the ancient authors' work. The uniqueness of the masterpieces comes from the poets' flexibility in the arrangement of words, as well as in the style of rhyming, although there are template rules in poetic forms that dictate the number of words contained in a sentence [8].

In the Vietnamese language, there are six more tones than the common rising and falling intonation of pitch [9], they are expressed orthographically by the diacritics including the grave (`), the acute accents (′), the hook above (̉), the tilda (~), and the dot under (.). Diacritics containing the grave and the unmarked are flat tone pitches, whereas the rest are uneven tone pitches [4], and they are strictly regulated by specific rules when creating in the form of the six-eight poetry.

Among the traditional Vietnamese poetry forms, the six-eight in which six and eight are the number of syllables in each line [8]. A simple six-eight creation has to adhere to the most fundamental principles: Considering a 4-line poetry (or a quatrain) example, and R_6^i is the rhythm of the last syllable in the i^{th} six-syllable sentence, $R_{8_1}^i$ and $R_{8_2}^j$ are correspondingly the 6th and 8th syllable of the j^{th} eight-syllable sentence, and X is the other syllables. Summarily, the rhyme system as presented below:

$$\text{X X X X X } R_6^1$$
$$\text{X X X X X } R_{8_1}^1 \text{ X } R_{8_2}^1$$
$$\text{X X X X X } R_6^2$$
$$\text{X X X X X } R_{8_1}^2 \text{ X } R_{8_2}^2$$

The rhyming flow through verses adheres to the rules that the R_6^1 is relevant to the $R_{8_1}^1$ rhythm, the $R_{8_2}^1$ rhythm is relevant to both R_6^2 and $R_{8_2}^1$. Additionally, if there is one more six-eight stanza (i.e., the third six and eight-syllable sentences), they must similarly fulfill the connection between the $R_{8_2}^2$ and the $R_{8_1}^3$ rhythm.

As mentioned previously, the two types of Vietnamese tone, which consist of flat and uneven tones, are also known as sharp and flat tones correspondingly [4], and the diacritics of both groups always combine with the vowels, not the consonants. Cases in point, the word "lá" (|la˧˥|) is a sharp-tone word, "ra" (|ra|) is a flat-tone word since the acute accent, the unmarked goes with the vowel 'a' sequentially, and the word "ngả" (|ŋaˀ˧˩|) belongs to the sharp group due to the hook above the letter 'a'. For every six-syllable verse, it is crucial to comply with the tone flow rules of "flat-sharp-flat" for the 2nd, 4th and 6th words, and for the eight-syllable, the rules of "flat-sharp-flat-flat" for the 2nd, 4th, 6th and 8th words must be satisfied. Hence, the making of a six-eight poetry masterpiece is a combination of lots of factors such as maintaining the fundamental template rules, soundly rhyming, semantics, uniqueness of content, as well as the acute of words using in proper order that depends on the poets' extensive knowledge. Nonetheless, Dale [10] and Khurana *et al*. [11] show that computers can understand the context themself by analyzing the relation between words directly on text sequences. There has been much work on enhancing the ability of keywords and texts extraction of computers, for instance, Mitrpanont and Chongcharoen [12] approached the problem of translating the Thai language to English by using disambiguation techniques, Horita and Kimura *et al*. [13] presented a system that automatically extracts the keywords of online documents in different languages (such as Japanese, Chinese, Korean, etc.) for the effective Wikification of them. For the work on texts, Georgiades and Andreou [14] presented their work of a novel software tool—NALASS that automates the Requirements Engineering process consisting of creating question sets for the elicitation, organizing in the analysis stage and generating graphical notations or a mobile application of Kushol and Ahsan *et al.* [15] that processes the texts coming from images (i.e., captured images of flyers, newspapers, books, etc.) that could help users to take note of information they need from those sources effectively.

Recently, researching natural language generation has been an engaging topic with many extraordinary literature creations made by computers [10, 11]. Namely, computationally generated Vietnamese poems with context control and rhyme constraints maintained throughout verses [16] or a Chinese poem generator with consistency in theme and flexibility of composing in diverse genres [17]. It is obvious that the necessity of maintaining the context should be satisfied since it keeps the entire creation to be meaningful, therefore, research in the field of literature generation requires algorithms that can analyze the relation of sequential data in the form of words in texts. Accordingly, there are a variety of proposed models that approach through RNN, Long-Short Term Memory algorithms, e.g., an English poetry generator implemented with an LSTM-based model that was trained on generic poems [18], or a program in which the algorithm automatically expands input keywords to phrases and connect them to form full poetry verses [19].

As outlined before, the model for Chinese poetry that can avoid theme-shift is trained not only on word connection, but also the sequence-to-sequence learning using an attention-based recurrent neural network [17]. And the other, which generates Vietnamese poetry, is an LSTM memory model, namely the SP-GPT2 (Semantic Poem—GPT2 model) [16]. However, most of them demand strong computational hardware and take a large amount of time, thus, our approach is to reduce the cost of time and make it easier for a moderate hardware but capable of making a simple semantic six-eight poem.

It is reasonable for our approach since there has not been much research on the topic of Vietnamese poetry, e.g., there is Nguyen *et al.* [16] GPT2 model-based with the requirement of a huge number of poems for the training. The entire work focuses on the development of the program, not the comparison between core-generating algorithms. Briefly, there are two main points: Firstly, the evaluation of experimental creation showed that the model gives good results with preservation in template despite the medium-sized dataset, and secondly, the computational poems are close to human-written work, which is proven by the scores of small blind-tests.

The rest of this paper consists of four sections: Section II contains the methods of composing Luc bat poetry, the RNN-based model, the refining algorithms, and the processing of data. Section III presents analyses of some generated poems, Section IV discusses the evaluations of generated poems and implementation on the service robot, and Section V summarizes our presented work.

II. The Methods

A. The Making of Six-Eight Poetry

Simile, comparison, metaphor, and metonymy are adapted by Vietnamese ancient poets for expressing the correlation between general objects and iconic images [20], thus, it requires a dataset with a wide range of topics from human beings' daily life to the surrounding nature, which is the reason for us to prepare paragraphs that contain Vietnamese cultural information, national customs, emotional expressing and so on (Fig. 1). These paragraphs help the model cover diverse topics as well as widen the words dictionary to make a turning point in the composed poem.

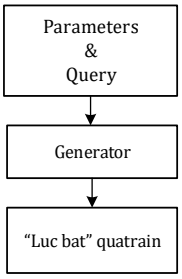

Figure 1. Summary of a poem generating process.

Trained parameters and input query are pre-condition in which the query can either be a sentence or simply a sequence of keywords relative to the chosen topics. For the generation process of a verse, every word is checked by the refining model and replaced whether itself did not adhere to the template rules before finally outputting a stanza. For theme-shift avoiding, the content of the first complete eight-syllable is extracted by choosing the words that describe the characteristics, motions, or changes, those words form a new input query to generate the next six-eight stanza. At last, a new six-eight quatrain is composed based on the input query context. The process of generating a six-eight quatrain is shown in Fig. 2.

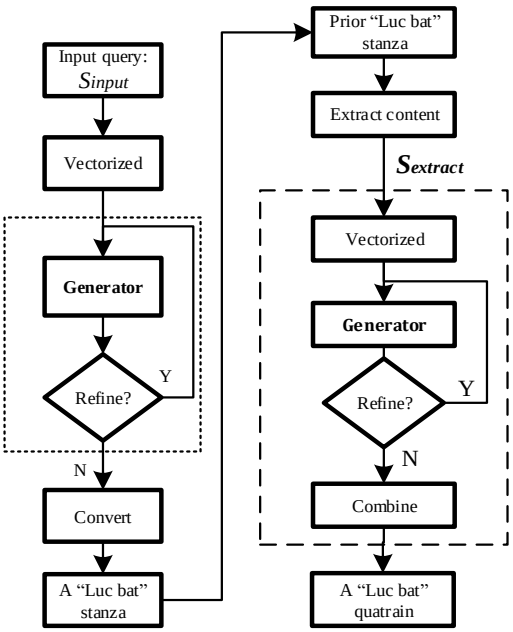

Figure 2. The process of generating a six-eight quatrain.

B. The RNN-Based Model and Refining Algorithms to Compose the Six-Eight Poetry

Beginning with an input sequence S_{input} in which syllables are separated and vectorized into K_{input} one-hot vectors denoted as x^{i}_{input}, where i is the index and has the value in the range of $[0, K_{input} - 1]$. Each vector has the size of $1 \times W$ (W is the number of words in the dictionary processed from the dataset).

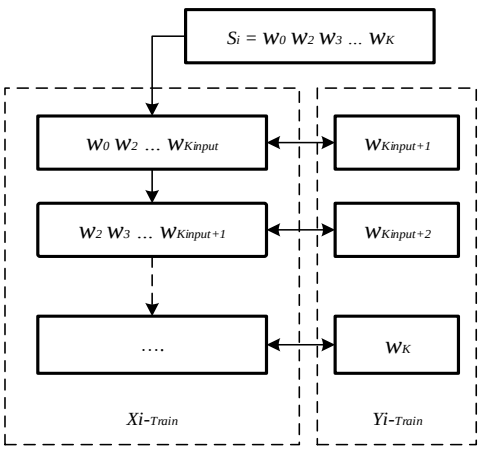

Figure 3. A process for every sequence S_i to distribute the i^{th} X_{Train} and Y_{Train}.

The set of vectors X_{input}, as illustrated in Fig. 3, is then sent to the RNN-based generator to compose a first new stanza, however, the output, which is another set of vectors rather than a couple of verses, must be converted into sentences by decoding from a one-hot to its corresponding words that are stored.

$$X_{input} = \left[x_{input}^{0} \quad x_{input}^{1} \quad \dots \quad x_{input}^{K_{input}-1} \right]^{T} \tag{1}$$

The generating process continues with the query extracted from the previous eight-syllable verse into the form of $S_{extract}$, which contains the number K_{input} of keywords, is similarly vectorized the same as the first generation into Eq. (2). The set is then fed to the generator to start the composing procedure again, as long as there is no mismatched syllable to the template rules, the two stanzas are combined to form a complete six-eight quatrain, as shown in Fig. 2.

$$X_{extract} = \left[x_{extract}^{0} \quad x_{extract}^{1} \quad \dots \quad x_{extract}^{K_{input}} \right]^{T} \tag{2}$$

C. Processing of the Dataset

By synthesizing from Vietnamese literature books, writings on many topics [20–26], a corpus set of 43,031 short paragraphs is distributed for training the model. However, they are raw materials with many disallowed characters that could eventually make the training to be complex, thus, they must be checked and refined to eliminate the disallowed before vectorizing process happens. The list of disallowed consists of the bracket of all types, the quotation marks (“), the colon (:), the left and right slashes (/), (\), the exclamation mark (!), the question mark (?), the hyphen (-) and the ellipsis (…). According to Fig. 2, the input query is a K_{input}-word sentence so every paragraph has the number of words that is equal to or more than K_{input} is allowed. At this stage, all the processed sequence is sliced into K_{input}-syllable. Table I below summarizes the number of paragraphs through stages, there are 23,095 paragraphs after the disallowed word amount has been checked, and a total of 3,000 chosen words that have high rates of presence itself overall of paragraphs.

TABLE I. NUMBER OF PARAGRAPHS THROUGHOUT STAGES

Raw Set	Passed	Chosen Word
43,031	23,095	3,000

Every sequence of S_i Eq. (3) is divided into two sets of X_{train} and Y_{train} in which the X_{train} contains K_{input} –word, whereas the Y_{train} value is the next corresponding word of that phrase. Nonetheless, there are two cases when processing an S_i that the number of K can be equal to or larger than K_{input} so for the case of equality, the Y_{train} will be the space character.

$$S_i = \{ w_0 \quad w_2 \quad \dots \quad w_K \} \tag{3}$$

where w_j (j from 0 to K) is the jth syllable of sentence S_i (i from 0 to 23,094).

In summary, Y_{Train} is a set of 1,559,196 words with respect to 1,559,196 phrases of the X_{Train} that are processed from 23,095 previously passed paragraphs.

D. The Recurrent Neural Network Based Generator

As shown in Fig. 2, the main generator bounded by the dashed lines contains two sub-block of “Generator” and “Refine”, these two are used to predict the next word to suit the poetry verse. It requires looping to fully make a six-eight stanza since every prediction only produces one word, so for each stanza, the demanded number is 14 syllables corresponding to 14 one-hot vectors. The embedded procedure of the “Generator” sub-block is explained in the pseudo-code of Algorithm 1 below.

Algorithm 1. Algorithm for generating a stanza

Generate(S_{in})
in: Input a K_{input}-keyword sequence:
$S_{in} = \{ w_{in}^{0}, w_{in}^{1}, ?\ \ w_{in}^{K_{input}-1} \}$
out: A sequence of S_{out}
temp: One-hot vector set A_{in}
Predicted vector v_{out}
Converted word w_{out}

$A_{in} \leftarrow Texts-to-Vectors-Encoding(S_{in})$
$S_{out} \,||\, S_{in}$
$i \leftarrow 0$
while $|S_{out}| < 14$ **do**
  $w_{out} \leftarrow$ *Predict-next*(A_{in})
  $S_{out} \leftarrow S_{out} \,||$ *Vector-Decoding*(w_{out})
  $i \leftarrow i+1$
  $S_{in\text{-}next} \leftarrow sub(S_{out}, i, i + K_{input} - 1)$
  $A_{in} \leftarrow$ *Texts-to-Vectors-Encoding* (S_{next})
end while
return S_{out}

Every input query S_{in} is vectorized into a one-hot set and assigned to the temporary array A_{in} that is used to predict the next value of v_{out}, which is then decoded to the word w_{out} and concatenated to the S_{out} sequence. The process gradually continues until the length of S_{out} has fully reached the number of 14 syllables (not considering the space character). In each looping step, the input for every new prediction is an extraction of the previously combined sequence that contains K_{input} syllables indexing from the i^{th} to the $(i+K_{input}-1)^{th}$, where i is a counter beginning from 0.

Algorithm 2. Algorithm for predicting

Predict-next (A_{in})
in: Array A_{in} of K_{input} one-hot vectors
out: Predicted vector v_{out}
temp: Output vector y has value ranging from 0 to 1

▷ Vector y is a predicted vector of the input array A_n
y ← g(A_{in}) ▷ g is the predicting function
$v_{out} \leftarrow$ *Classify*(y)
▷ The *“Classify”* function is a combination of the sigmoid and softmax function
return v_{out}

The pseudo-code above shows the process of predicting a vector output vout, an input array A_{in} is fed to *g* which is an RNN algorithm-based function to predict a new vector of *y*. However, this value of *y* cannot be directly used to decode since its values range from 0 to 1, which must be first classified as only 0 or 1. The classifier is implemented with the function of sigmoid and softmax which are presented in the next section.

E. Recurrent Neural Network Algorithm

A Recurrent Neural Network algorithm is considered to deal with sequential or time series data in which every data point is relative to its previous [27] accordingly, we built an RNN-based algorithm that has two types: the many-to-many (left dashed line bounded group) and the many-to-one (right dashed line bounded group). In summary, the entire generating function *g* (pseudo code of Algorithm 2) has three main parts that are the two recurrent layers with a drop-out layer between them. For the many-to-many layer, it receives a vector set of the input query and outputs extracted features of each word token, which are then fed to the drop-out for regularization by randomly zeroing each hidden node throughout the training phase since this help avoid the overfitting of a neural network [28]. And lastly, the many-to-one performs as the main predictor in which the values distributed by the drop-out are the inputs for calculating the vector $\hat{y}_{output}$ (Fig. 4).

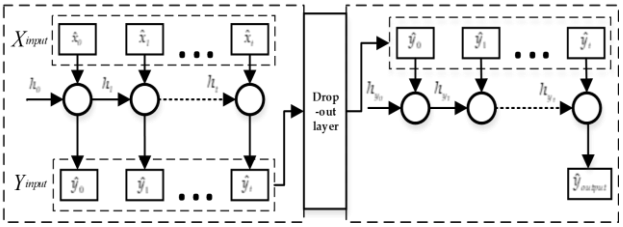

Figure 4. Architecture of RNN-based algorithm.

Every $\hat{y}_t$ is computed as Eq. (4) with subscript *t* denoting the time step:

$$\hat{y}_t = f_1(h_t) \tag{4}$$

$$h_t = f_2(x_t, h_{t-1}) = f_2\left(w_{x_t} \times x_t + w_{h_{t-1}} \times h_{t-1} + b_h\right) \tag{5}$$

The value of w_{x_t}, w_{x_t}, $w_{h_{t-1}}$ are the weights associated with the activation x_t, $h_{t-1}(t \varepsilon 1)$ and b_n is the bias value of the first recurrent layer. Each h_t is calculated by the function of f_2 with the inputs of x_t, the previous layer activation value $h_{t-1}(t \varepsilon 1)$ with corresponding weights *w*, and biases as Eq. (5). It is found that every new activation involves both previous state information ($h_{t-1}(t \varepsilon 1)$) and present inputs $\hat{x}_t$ .Hence, each h_t is a memory element of the X_{input} [27].

The vector set $Y^{input} \in R^{K_{input} \times W_{mid}}$, where W_{mid} is number of columns after the first layer, is used to compute the final y_{output} as Eq. (6) in which each activation h_{y_t} (*t* denotes the timestep) is calculated as Eq. (7).

$$y_{output} = f_1\left(h_{y_t}\right) \tag{6}$$

$$h_{y_t} = f_2\left(w_{y_t} \times y_t + w_{y_{t-1}} \times h_{y_{t-1}} + b_{h_y}\right)(t \geq 1) \tag{7}$$

where w_{y_t}, $w_{h_{y_t-1}}$ are weights associated with vector y_t, activation h_{y_t-1} and b_{h_y} is the bias for the second recurrent layer. Since then, the loss function *L* of $y^{(t)}$ is computed based on the loss value as the backpropagation is conducted at every time step and mathematically presented in Eq. (8) [27]. At a particular timestep *T*, the derivation of loss *L* concerning the weight *x* is shown in Eq. (9):

$$L(\hat{y}, y) = \sum_{t=1}^{T} L\left(\hat{y}^{\langle t \rangle}, y^{\langle t \rangle}\right) \tag{8}$$

$$\frac{\partial L^{(T)}}{\partial x} = \sum_{t=1}^{T} \frac{\partial L^{(T)}}{\partial x}\bigg|_{(t)} \tag{9}$$

Finally, for classifying values of the y_{output}, two dense layers are applied with the function of sigmoid and softmax that output the value of only 0 or 1. The mathematical form of the two functions is below.

$$f(x) = \frac{1}{1+e^{-x}} \tag{10}$$

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{11}$$

where $\vec{z}$ is an input vector with each element in form of z_i.

F. The Refining Algorithms

As illustrated in Fig. 5, there are two cases of the verse's type: if it is a 6-syllable, the checking rule is "flat-sharp-flat" and otherwise, it must be "flat-sharp-flat-flat". For every passed sentence will be directly transferred to the next rhythm-checking step, shown in Fig. 6 below. And for any rule-mismatched word, it will be automatically replaced with a synonym that can meet the template rules. The replacing process sequentially runs from the minimum faulty syllable index, namely, the 2nd syllable would be replaced first, if it didn't adhere to the rule. Nonetheless, there is an issue of which a mismatched word may have more than one synonym that can suit the empty slot. Hence, the algorithm must be implemented with a synonym-choosing algorithm that picks the fittest one to replace the mismatched tone as well as relate to the content.

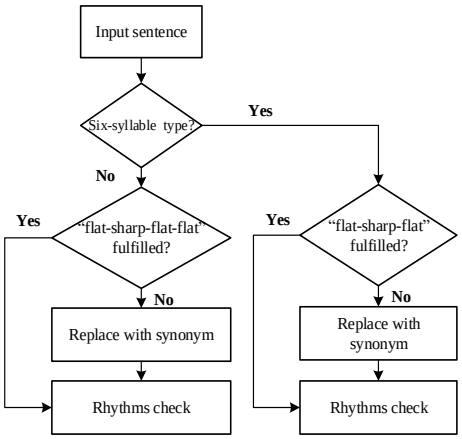

Figure 5. Tone-checking algorithm.

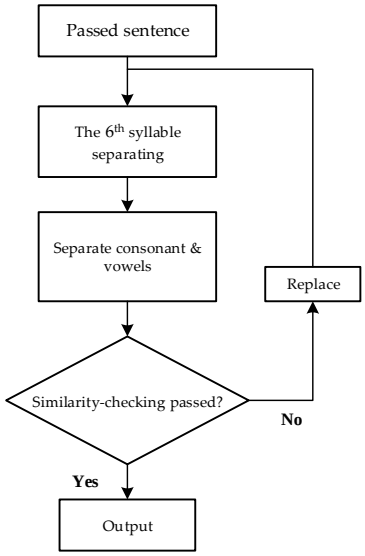

Figure 6. The process of checking the rhythms.

The rhythm-checking process extracts the 6th words from both the six and eight-syllable sentences, and the diphthongs of them are separated to compare for similarity with the rules that if they were the same, the algorithm would end up the refining process, otherwise, it would have replaced with a suitable. However, there is an issue with choosing the word to fit since there could be only one or both two of them that have been replaced in the tone-checking process. For the first case, the replaced one is chosen to retain the content of the unchanged, and for the remaining case, the word of the eight-syllable sentence is chosen due to the six-syllable is the meaning root of a stanza so it should remain.

III. Results

Hyperbolic tangent activation Eq. (12) is implemented for both functions of f_1 and f_2 to generate two different six-eight poems, namely the poem (A) and (B).

$$tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{12}$$

For the size of the input vector ($K_{input} \times W$), it is set to the value of 6×3000, and the dictionary has a total of 3000 words. As a result, two poems are presented in Figs. 7 and 8 with the International Phonetic Alphabet (IPA) transcriptions of target syllables (mentioned in "The Refining Algorithms") that are taken note of below each poetry verse as in Sherstinsky [27]. Moreover, all the syllables go with its superscript denoting the type of tone, where "F" and "Sh" are the abbreviations for "flat tone" and "sharp tone" correspondingly. The experiments of this stage are directly conducted on the program, not the communication between the robot and human participants which will be shown in the next section.

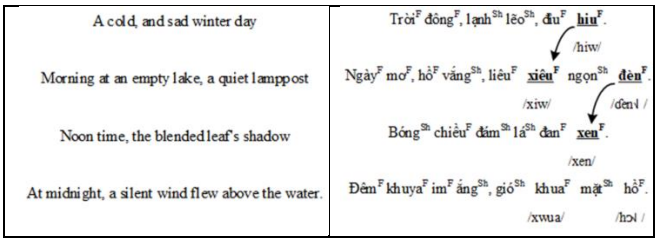

Figure 7. Generated six-eight poem namely (A).

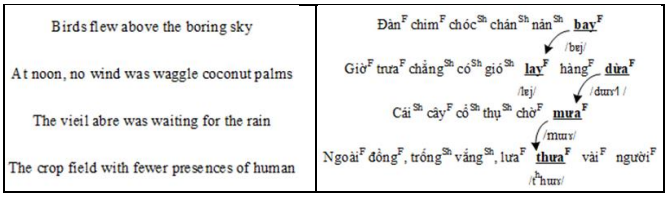

Figure 8. Generated six-eight poem namely (B).

Beginning with the first poem (A), an input query consists of {"mùa đông" (the winter), "vắng tanh" (empty)}, "lạnh lẽo" (cold)} as the keywords for the creation. It is observed that the tone flow meets the first condition, however, there are mismatched diphthongs of the two syllables: "xen" and "khua" since the transcription $|e|$ is different from the $|wua|$. This weird point is caused by stopping the refining process because there had been a repeat run of the program to find the fittest word for the position without any endpoints. The small size of the dictionary is the main reason for this result.

The next poem (B) follows the template rules better than the poem (A), accordingly, R_6^1 and $R_{8_1}^1$ matched with sound of $|ɐj|$ and there are triple diphthongs-matching of $R_{8_2}^1$, R_6^2, and $R_{8_1}^2$, which are the sounds $|ɯɤ|$. In the case of poem (B), the keywords set of {"đàn chim" (the birds), "bay lượn" (flying), "chán nản" (boring)} that guides the quatrain's theme to describe a noon time that has no rain, wind or presents of human, the scene is boring with only some birds flying above the sky whereas coconut palms are standing quietly. There was a turning point in this creation which is the unexpected appearance of a vieil arbre ("Cây cổ thụ") as the main object of the third poetry verse. Summarily, two poems show that the program can maintain the fundamental rules of Vietnamese six-eight poetry as the generated results are not only semantic but also creative.

IV. Discussion

A. Human Evaluation

Due to the ineffectiveness of the BLEU score in showing how great a context of a generated poem involves, thus, it is unnecessary to evaluate the poem task as Nguyen *et al*. [16] and Zhang and Lapata [19] have shown. Accordingly, four criterias are recommended in Table II. A survey was conducted to assess the opinions of participants, who are college students, lecturers, and literature students, there were two sets of poems consisting of the computationally generated and the human-written that were scored within the range from 1 to 5. Participants without knowing the type, are required to read and mark all poems of both sets.

TABLE II. Evaluating Criteria and Their Definitions

Criteria	Definition
Compliance	Showing whether a poem fulfills template rules
Coherence	Showing how conherent a poem is
Content	Indicating the context throughout verses
Fluency	Showing how smooth the verses are rhymed

TABLE III. Human Evaluation Scores

Criteria	Machine-created	Human-written
Compliance	3.64	3.78
Coherence	3.43	3.65
Content	3.60	3.88
Fluency	3.32	3.81

According to Table III, the program has close scores of the "Compliance" and "Content" criteria to the human-written, however, by the lower scores of "Fluency", as well as "Coherence", it can be stated that the generated results haven't over-passed the creations of human and strongly impressed on reviewers.

B. A Blind Test

The previous survey can be considered as a small blind test with the true marks that reflect the opinions of reviewers, but the case of this blind test, which is a Turing test-based evaluation, only requires participants to distinguish whether a particular poem is a creation of human or computationally composed.

As shown in Table IV, a percentage of 56.75, which indicates the number of wrongly identified Computationally-Composed (CC) poems as the Human-Written (HW) type, is quite a turning result since the lower scores of them than the human-composed ones in the criteria of "Fluency" and "Coherence" shown in Table III have partly shown that readers may easily to find out a poem true origin through defining its context connection between verses.

TABLE IV. Blind Test Scores

CC as HW	HW as HW	Undefinable
44%	43.25%	12.35%

Moreover, the ratio of undefinable cases is expected to be smaller than 5% due to the participants mostly majoring in Literature, nonetheless, the capability of quite strictly following the template rules, as well as the high computational creativity have confused the readers. Thus, it is evident to consider the good performance of the program in generating Vietnamese six-eight poetry.

C. Service Robots Implementation

As mentioned previously, the implementation of service robots is conducted with the participation of MIA—a receptionist robot that helps users to look up information about an office building. MIA is able to sing, talk, and make a handshake to its first met person as a polite behavior. It is now enhanced with a more ability to compose a six-eight poem that bases on the user's emotions, weather conditions, or simply a friendly greeting (Fig. 9).

(a) (b)

Figure 9. MIA robot—a receptionist robot with its ability to greet with poems. (a) A friendly handshake of MIA the Robot. (b) A greeting with Vietnamese traditional six-eight poetry makes the first impression.

The guests were strongly impressed by the ability to interact with both politeness and Vietnamese national identity. This makes them feel relative to the receptionist robot from the first meeting rather than the few available sentences only for some specific cases of providing information.

D. Compare to Other Approaches

The most possible similar research on Vietnamese poems generator would be Nguyen *et al*.'s work that presented the SP—GPT2 model to generate Vietnamese Luc bat poems [16]. Their model was trained with over 2 million Luc bat sentences with a more complex model structure than ours, which was trained with 23,095 sentences. The two approaches are based on the use of Recurrent Neural Networks (RNN). However, the big difference between the two approaches is the method of maintaining the template rules, which is the implementation of refining algorithms in our work, and the contextual vectors model of Nguyen *et al*. [16]. We aim to construct a model requiring less hardware power and time to train but can make poems that are closer to the human-written ones. Thus, we presented an approach of using refining algorithms to fulfill template rules.

V. Conclusion

The study presented an implementation of a recurrent neural network-based program with refining algorithms that enables a service robot to compose Vietnamese six-eight poetry. The diversity in the theme of the generated results with their good adherence to the poetry template rules has enhanced the human-robot verbal interaction with a strong impression and interest in users. Last but not least, the turning points in the human evaluation and the blind test show that the results are close to the human written ones since the misdistinguished percent is over 50, and quite a high rate of undefinable. For the development of future work, the model used for the generator could be replaced with other generative models to find one that makes meaningful, rhythmic, as well as diverse themes. Or this model could be trained with a larger dataset containing more poetry combinations and sentences to enhance the output poems, or even reduce the gap between human-written and machine-generated masterpieces.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Nguyen Hai Dang—wring original draft, visualization, validation, configured, coding. Nguyen Truong Thinh—methodology, writing, review and editing, project administration. Both authors had approved the final version.

Funding

This research is funded by University of Economics Ho Chi Minh City—UEH University, Vietnam.

References

[1] N. Mavridis, "A review of verbal and non-verbal human-robot interactive communication," *Robotics and Autonomous Systems*, vol. 63, pp. 22–35, 2015.
[2] M. Makatchev *et al.*, "Dialogue patterns of an Arabic robot receptionist," in *Proc. 2010 the 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, IEEE, 2010.
[3] H. Ishiguro *et al.*, "Robovie: An interactive humanoid robot," *Industrial Robot: An International Journal*, vol. 28, no. 6, pp. 498–504, 2001.
[4] N. Mitsunaga *et al.*, "Robovie-IV: A communication robot interacting with people daily in an office," in *Proc. 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2006.
[5] K. Ogawa, K. Taura, and H. Ishiguro, "Possibilities of androids as poetry-reciting agent," in *Proc. 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, IEEE, 2012.
[6] N. Du, *The Tale of Kieu*, Yale Univerity Press, 1973
[7] M. Ross, "DU, (Thong, trans. and annot.), 'The Tale of Kieu' (book review)," *Journal of Asian Studies*, vol. 33, no. 4, p. 743, 1974.
[8] A. H. Pham, "Poetry translation from a tonal language (Vietnamese) to a non-tonal language," *Delos: A Journal of Translation and World Literature*, vol. 35, no. 2, pp. 140–128, 2020.
[9] G. H. Emerich, "The Vietnamese vowel system," Doctor dissertation, University of Pennsylvania, 2012.
[10] R. Dale, "Natural language generation: The commercial state of the art in 2020," *Natural Language Engineering*, vol. 26, no. 4, pp. 481–487, 2020.
[11] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimedia Tools and Applications*, pp. 1–32, 2022.
[12] J. L. Mitrpanont and P. Chongcharoen, "TH_WSD: Thai word sense disambiguation using cross-language knowledge sources approach," *International Journal of Computer Theory and Engineering*, vol. 7, no. 6, pp. 428–433, 2015.
[13] K. Horita, F. Kimura, and A. Maeda, "Automatic keyword extraction for Wikification of east Asian language documents," *International Journal of Computer Theory and Engineering*, vol. 8, no. 1, pp. 32–35, 2016.
[14] M. G. Georgiades and A. S. Andreou, "A novel software tool for supporting and automating the requirements engineering process with the use of natural language," *International Journal of Computer Theory and Engineering*, vol. 5, no. 3, pp. 443–449, 2013.
[15] R. Kushol, I. Ahsan, and M. N. Raihan, "An android-based useful text extraction framework using image and natural language processing," *International Journal of Computer Theory and Engineering*, vol. 10, no. 3, pp. 77–83, 2018.
[16] T. Nguyen, P. Nguyen, H. Pham, T. Bui, T. Nguyen, and D. Luong, "SP-GPT2: Semantics improvement in vietnamese poetry generation," in *Proc. 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2022.
[17] Q. Wang, T. Luo, and D. Wang, "Can machine generate traditional Chinese poetry? a Feigenbaum test," in *Proc. 8th International Conference on Brain Inspired Cognitive Systems*, 2016
[18] J. Hopkins and D. Kiela, "Automatically generating rhythmic verse with neural networks," in *Proc. the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 168–178.
[19] X. Zhang and M. Lapata, "Chinese poetry generation with recurrent neural networks," in *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 680–670.
[20] T. V. Thien, "The vitality of Luc bat in Vietnammese modern poetry," University of Khanh Hoa, 2019. Available: https://ukh.edu.vn/chi-tiet-tin/id/1461/Suc-song-cua-the-tho-Luc-bat-trong-tho-ca-Viet-Nam-hien-dai (in Vietnamese)
[21] H. Thanh and H. Chan. (2018). Vietnamese poet (Thi nhân Việt Nam). [Online]. Available: https://gacsach.online/doc-online/thi-nhan-viet-nam-hoai-thanh-hoai-chan.html (in Vietnamese)
[22] N. Duy, "Poetry of Nguyen Duy (Thơ Nguyễn Duy)," Writers Association Publisher, 2010. (in Vietnamese)
[23] T. Huu. (1962). North Vietnam (Việt Bắc). [Online]. Available: https://loigiaihay.com/viet-bac-to-huu-c360a51380.html (in Vietnamese)
[24] T. Da. (1989). Tang Poetry (Thơ Đường). Young Publisher: 240-1. [Online]. Available: https://vietbooks.info/threads/tho-duong-tan-da-nxb-tre-1989-nguyen-quang-tuan-250-trang.85465/ (in Vietnamese)
[25] N. H. Thiep. (2005). Throw a net to catch birds. [Online]. Available: http://nguyenhuythiep.free.fr/giangluoi/index.html (in Vietnamese)
[26] P. Ngoc. (1985). Learn about Nguyen Du's artistic style in Kieu stoty, Social Science Publishing House, Hanoi. (in Vietnamese)
[27] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, 132306, 2020.
[28] J. Choe and H. Shim, "Attention-based dropout layer for weakly supervised object localization," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.