International Journal of Engineering

Journal Homepage: www.ije.ir

Determining the Composition Functions of Persian Non-standard Sentences in Terminology using a Deep Learning Fuzzy Neural Network Model

H. Motameni*

Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran

P A P E R I N F O

Paper history:
Received 01 August 2020
Received in revised form 02 September 2020
Accepted 03 September 2020

Keywords:
Nonstandard Sentences
Recurrent Neural Network
Standard Sentences

A B S T R A C T

Organizations can enhance the speed of well-informed decision-making by correctly understanding and using data. Since there is a tremendous gap between the speed of data processing and data generation in the world, exploring data mining in the digital world becomes inevitable. In the Persian language, similar to other languages, with the expansion of communications through social networks, the spelling of words has become abridged and the engagement of foreign loan words and emoticons has been increasing on a daily basis. Given the richness of Persian and its typographical-grammatical similarities to Arabic, research in Persian can be applied to other akin languages as well. In this regard, the current study deals with data mining of Persian non-standard sentences in order to find the function of each word in the sentence. The volume of computation might be limited in traditional methods of natural language processing for each factor contributing to functions. That is because the minimum number of computations is (5 × number of words 9) + (5 × number of words 15). Therefore, this study adopted the Gated Recurrent Unit (GRU) method to process such computations. The newly proposed method reinforces the results of word function identification by using two categories of "independent" and "dependent" Persian language functions as well as five factors contributing to the functions of words in sentences as five output gates. Meanwhile, the values of the training tables in this method are fuzzy, where the center-of-gravity fuzzy method is adopted to decide on the fuzzy values as well as to reduce the complexity and ambiguity of such computations on the probability of each event occurring. Therefore, the new method is briefly called "fuzzy GRU". The results show that the proposed algorithm achieves 80 % reduction in the amount of calculations per gate of updates and reinforcement is approximately 2 % up from 67 % in standard sentences to 69 % of the non-standard sentences.

***doi**: 10.5829/ije.2020.33.12c.06*

1. INTRODUCTION

Also known as Farsi, Persian is spoken officially in several countries including Iran, Afghanistan and Tajikistan. In addition, it is used in a few countries as a second language. Moreover, Persian was remarkably predominant in other Asian and European languages. This language is very much akin to Arabic in terms of alphabet and grammar. These issues have compelled scholars to more than ever explore the Persian language [1]. In the modern age, the expansion of social networks among the public has led to divergence of writing and speech styles in every language, turning phrases into a rather abridged, colloquial form. Therefore, the current study has covered nonstandard sentences of Persian language. One of the most important reasons behind doing such research is that the results are exclusively applicable to a number of linguistic areas such as smart filtering [2], machine translation [3], speech recognition [4, 5], text recognition [6], text summarization [7], etc. The majority of Persian linguistic research projects adopt traditional methods of neural networks. In this regard, our study has used this system to resolve the defects in the previous methods [8, 9] and to benefit from the disambiguation property of fuzzy neural networks. In the newly proposed fuzzy system, center of gravity defuzzification has been employed owing to its versatility and practical advantages [10, 11]. This system

*Corresponding Author Institutional Email: *motameni@iausari.ac.ir* (H. Motameni)

Please cite this article as: H. Motameni, Determining the Composition Functions of Persian Non-standard Sentences in Terminology using a Deep Learning Fuzzy Neural Network Model, International Journal of Engineering, Transactions C: Aspects, Vol. 33, No. 12, (2020), 2471-2481.

relies on Gated recurrent units (GRU) architecture [11] with fuzzy values, because it simply uses 0 and 1, while there are fuzzy training matrix values for deep learning recurrent neural network so that accurate decisions can be made in values [0, 1]. In the recovery gateway section, after sorting in ascending order the values obtained in each section, 80% of the bottom values are discarded and the top 20% are transferred to the next class.

To solve the above problems, this paper makes four contributions:

- The output of this research can be used in all data mining projects mentioned above.
- The method of this research can be used in all Persian-speaking countries and in Arabic-speaking countries.
- The calculation accuracy in the proposed method is 2% higher in standard sentences than in non-standard sentences.
- The proposed method in terms of computational complexity at each stage of the transition from each of the five factors affecting the acceptance of the role is reduced by 80%.

2. LITERATURE REVIEW

This section first delves into linguistics as the original discipline, and then discusses computational linguistics and morphology as two major sub disciplines. The next section describes the new fuzzy system adopted in this paper, while reviewing the relevant literature on fuzzy techniques for Persian. Since the system proposed in this paper is a classified fuzzy system, we will next explain why it has been named so, and generally elaborate on the independent and dependent roles. Finally, nonstandard sentences and words are defined.

2. 1. Linguistics Language is a complex phenomenon, where any precise and comprehensive investigation on a language require knowledge from numerous fields, including sociolinguistics, psycholinguistics, neurolinguistics, forensic linguistics, clinical linguistics, analytical linguistics, educational linguistics, logic, and even computer sciences over the last few decades. Linguistics include the fields of grammar, syntax, phonetics, phonology, semantics, pragmatics, discourse analysis, comparative historical linguistics and typology, reflecting its various dimensions [12, 13]. The definition of linguistics states: "a science that systematically studies language". Thus, a linguist is the one who "conducts linguistic studies" [14].

Studies in the field of linguistics date back several centuries, but the linguistics research in its modern sense is totally recent, barely stretching back to a hundred years [14]. In fact, early studies on language were written in Sanskrit grammar by Indian Pāṇini during the fifth century BC. Later on, William Jones, a British lawyer, conducted significant research into linguistics. Ferdinand de Saussure, a Swedish linguist, established structuralism in linguistics over the first half of the twentieth century [15]. Noam Chomsky (1957) considered transformational grammar as a technique to examine language syntax, where "sentence" is a unit of study for linguistics. Chomsky's linguistic ideas are still popular in North America today. In the 1960s, a rival method to that of Chomsky was introduced with a discourse approach, where sentence is not studied as an independent unit, but as a dependent element within a context. Every text contains three semantic levels: "What is content about?", "How does interaction take place in the Equation of this content?", and the third level examines to what extent sentence (as a textual element) is helpful in formulation of content. This method even views word as text. Hence, text is not restricted to a pre-specified length [15]. Regarding the Persian language, the early research was carried out at the Department of General Linguistics and Ancient Languages, Faculty of Literature and Humanities, Tehran University [15]. The first Persian grammar was developed by the Iranian Linguistics Foundation. In the final years of the 20th century, Bateni conducted a series of studies obtaining a variety of sentence structures through the Persian grammar. He then investigated how each sentence could be converted into other types.

2. 1. 1. Computational Linguistics One of the most fascinating branches of linguistics is known as computational linguistics, which dates back no longer than fifty years. In a definition provided by Dr. Meghdari, computational linguistics refers to an interdisciplinary field consisting of linguistics and computer science, serving to model natural language through statistical and rule-based techniques for machine use [16, 17]. Computational linguistics initially covered only machine translation. In fact, many researchers sought after machine translation from the earliest days of the advent of computers.

This avenue of research was initiated in the 1950s. The first specialized journal on computational linguistics was known as Mechanical Translation published in 1954. Later on, the Association for Computational Linguistics was founded in 1962. Within a few years, the journal's title was revised into Computational Linguistics [18]. Nowadays, computational linguistics is applied in numerous fields and is not limited to machine translation [19, 20].

With respect to computational linguistics in Persian, several scholars such as Dr. Bijankhan and Dr. Shamsfard [21, 22] conducted exclusive research into word formation and construction of machine translators mentioned in [8, 23]. In addition, Iranian universities have taken a giant step in the Persian computational linguistics by admitting new students for this field in

recent years, while establishing several major computational linguistic labs. These labs include Web Technology Laboratory at Ferdowsi University of Mashhad [24], Institute of Humanities and Cultural Studies, Linguistic Research Institute, and the Center for Languages and Linguistics at Sharif University of Technology [16, 19]. Table 1 compares the most important studies in the field of Persian morphology.

In Table 1, the first two rows indicate the earliest and most important researches in the field of Persian corpora construction. In Oxford Dictionary, fuzzy has been defined as "having a frizzy texture or appearance, difficult to perceive; indistinct or vague". In another definition, "Fuzzy systems describe vague, inaccurate, and uncertain phenomena [25]," but this does not mean that the theory is inaccurate; on the contrary, fuzzy theory itself is a precise one. Introduced by Lotfizadeh, the fuzzy system then found its way into Persian linguistics research. It has been used in a limited way and often as a combination with Arabic language. In another study [8, 26], the fuzzy method was adopted to identify composition roles in Persian sentences.

2. 1. 2. Morphology Morphology has been given different definitions. One of the ordinary definitions is "the hybrid study of morphemes and their functions in words". Morphology involves specific steps, because grammar, alphabet, phonemes and speech vary in each language. For instance, morphology in English is different from those in Persian and Arabic [27–29].

2. 2. Recurrent Neural Networks Since, a basis of natural language processing is modeling the language; Recursive neural networks are a method to obtain a model of natural languages [30]. Recursive neural networks are a type of neural network. This type of neural network was presented in 1970, in studies LSTM recurrent neural network method peaked again and today is widely used in the processing of data series. Including these series there are writing, speech, text, meteorological data, etc. that have time series. The reason this neural network is recursive is that an operation is repeated on each series unit. In language processing, these units can be sentence, word, letters, etc. Figure 1 illustrates the recurrent neural network. There exists two traditional return networks namely, LSTM and GRU.

The reason for choosing LSTM method instead of GRU in this research is that the LSTM method in long sentences may be forgotten therefore, the GRU method was used to employ its long-term memory.

In the processing of any language, all input units from the first to the last one affect the results. In methods like LSTM, however, if the sequence is long enough, they will be "forgotten" and can only store the last few inputs in their memory. This problem is rooted in the inability of many traditional conditional methods. In 2014, Chou et al. proposed the GRU Recurrent Neural Network Method, which solved the problem of traditional methods with its long-term memory [11].

2. 3. Independent and Dependent Roles The Persian sentence roles include two independent and dependent categories. The independent roles include subject, predicate, object, complement and verb. The dependent roles include noun, adjective, genitive, governing genitive, dependent adverb, apposition, governing transducer, bending, retroactive exclamation and annunciator [14, 31]. Independent roles in Persian deal with the position of words in every sentence. As the name suggests, each independent role is applied without any dependence on another role. For this reason, we call these roles independent in Persian language sentences. Independent roles are also known as *primary* because they adopt the main position of words in Persian sentences [31]. Dependent roles are called so because they are applied in pairs. In this category of roles, the dependent pairs in a Persian sentence include noun-adjective, genitive-governing genitive, dependent adverb-verb/other role/sentence, apposition-governing transducer, bending-retroactive, and exclamation-annunciator. This structure, however, is not always true, since a sentence may show only one pair of dependent

TABLE 1. Comparison of morphological research

Researcher	Methodology	Advantages	Disadvantages
Bijankhan el al. [21]	Eagles standard	One of the basic Persian morphological corpora	Tagging words and constructing corpora merely based on word types, nonstandard corpora textual documents, eagles-based methodology
Assi and Abdolhosseini [23]	Manual	One of the basic Persian colloquial morphological corpora	Tagging words and constructing corpora merely based on word types, nonstandard corpora textual documents, manual methodology
Motameni and Peykar [8]	Fuzzy HMM	Using fuzzy system to determine word roles	Tagging standard sentences, high computational complexity
Motameni et al. [25]	Classified fuzzy	Low computational complexity, determining word roles, using fuzzy system	Only tagging standard sentences

roles. A few of paired roles are more frequently seen than others. Therefore, this paper divided the dependent roles into two subcategories: 1) dominant (noun, adjective, genitive, governing genitive and dependent adverb), and 2) rare (apposition, governing transducer, focused, exclamation, and annunciator) [31].

2. 4. Nonstandard Sentences and Words Given the increasing popularity of media and social networks, textual materials have been truncated or mistyped, sometimes transforming the formally established version of writing styles. It has therefore become critical to process such sentences in morphological applications. Generally speaking, there are several types of nonstandard sentences in every language.

1- Sentences with truncated words
2- Sentences with incorrect grammar
3- Sentences with loan words
4- Sentences with slang terms and emoticons,

which are senseless in the standard form.

This paper intended to investigate nonstandard sentences in 72 differently formulated sentences, while achieving the proper combination with the input sentence through classified fuzzy method [26].

3. PROPOSED METHOD

This study examines five different effective linguistic units adopting the role of words in the sequence of words and sentences. In each unit, specific fuzzy computations are frequently implemented on the sequences. In order to implement research in that procedure (i.e. a sequence of input units with variable lengths), the best possible method involves deep learning recurrent neural network. Each deep learning neural network has three gates (forget gate, update gate/input gate and output gate). One advantage of this method is that processing load is lower than that of in the method proposed in [8]. The new method proposed in this paper curtails the extent of computations in each class while enhancing the detection rate. By complying with the order of importance for each word adoption factor, it also improves the detection quality of word roles. Therefore, the novelty of the newly proposed method is that the fuzzy method is classified and the success rate has increased. In this paper, we first extract the two-dimensional arrays or matrices required for processing. Then, considering the success rate in finding the roles, the best combination of fuzzy classification is extracted in each array. In each of these classes, the results of possible role adoptions are arranged in ascending order. Then, 20% of the cases with the largest values are transferred to the next step, while the bottoms 80% are removed.

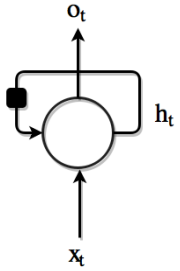

Figure 1. Recurrent neural network loop

Initially, we examined five major matrices processed in different fuzzy classes separately in 72 types of sentence structures. Then, we obtained the best possible composition. Section 2.3 provides 72 types of Persian sentences with different grammatical compositions of nonstandard sentences. In addition to the input sentences, their decompositions were imported into the system. Hence, the sentence decomposition was conducted through NLP Tools v 1.3.3 [24] or other similar software programs. In matrices dealing with alphabetical letters, a total of 44 Persian characters were covered In matrix dimensions, 21 represents the number of roles, i.e. 18 independent and dependent roles and 3 spacing characters. In addition, 10 in the matrix dimension indicates seven Persian word types plus three spacing characters.

3. 1. Sentence Formulation Scenarios In independent roles, there are $9^{\text{count of words in the input sentence}}$ number of possible composition scenarios. In this regard, the 9 represents the number of independent roles and spacing characters in Persian sentences (verb, letter, subject, predicate, object, complement and spacing character).

In dependent roles, there are $15^{count\ of\ words\ in\ the\ input\ sentence}$ number of possible composition scenarios. In this regard, the 15 indicates the number of dependent roles and spacing characters in Persian sentences (adjective, noun, adverb, unknown, apposition, governing transducer, bending, retroactive exclamation and annunciator).

3. 2. Training Matrices This section introduces the five main training matrices used to make decisions in the newly proposed method. Each of these arrays is obtained from 194 training sentences containing 76,274 words.

These five matrices have been summarized below:

A) Additional roles appearing after each role in sentences. Bi_gram_Combine: This is a two-dimensional matrix with 21x21 elements. This matrix determines the probability of each role occurring after another role (Bi-gram) in 194 Persian sentences.

$$1 \le Bi\_gram\_combine_{l.l+1}(k.j) \ge 0.1 \le k \le 21.1 \le j \le 21 \quad (1)$$

$$Bi\_gram\_Combine(k.j) = \prod_{i=1}^{count\ of\ (j)} Bi\_gram\_Combine_{i.i+1}(k.j)$$

In Equation (1), *i* indicates the word counter, *K* is the row counter, while *j* is the column counter. The value of Bi_gram for each word is a number between zero and one.

B) Composition role following each type of decomposition Transfer_Bi_gram: This matrix also functions as a transition from decomposition into composition. For the purpose of decomposition and composition, a total of 194 types of training sentences were tagged manually by experts. At the next step, data was inserted into a 10×21 table.

$$\text{Multiplication of jth roles placed after the ith type} \geq 0.1 \leq i \leq 10\ .1 \leq j \leq 21 \quad (2)$$

$$Transfer\_Bi\_gram(i,j) = \prod_{i=1}^{\text{count of (j)}} \text{Multiplication of jth roles placed after the ith type}$$

In Equation (2), *i* indicates the word type counter, while *j* indicates the word role counter.

C) Composition role in each type of decomposition (Transfer_Uni_gram): This can be considered as transition matrix, because it delivers the moment of transition (instead of word role in composition applied in each word type in decomposition). This is a 21×10 matrix. This 21×10 matrix can be obtained in 194 training sentences by tagging the decomposition and the composition values. Then, the production of multiplication indicates the mean of values as in Equation (3).

$$\text{Average value of jth roles replacing the jth type} \geq 0.1 \leq i \leq 10\ .1 \leq j \leq 21 \quad (3)$$

$$Transfer\_Uni\_gram(i.j) = \prod_{l=1}^{\text{count of (j)}} \text{Average value of jth roles replacing the jth type}$$

where, *i* is a word position counter, while *j* is the word role counter in the two-dimensional matrix.

D) Word-forming letters (Len_Word): The words weight matrix was obtained according to the letters in input sentences. These matrices contain 21×number_of_words elements, which is why we applied bi_gram tagging. Firstly, 76274 words are checked to see what character has followed another character in what percentage of words for each role. Divided by the total number, a numerical value from zero to one is obtained. Hence, Len_Bi_gram is extracted for each role, where the 44x44 array is a cell as large as the characters under examination. After obtaining the Len_Bi_gram matrix for each role with regard to the input sentence, the Len_Word matrix is extracted using Equation (4).

$$\text{Len\_Bi\_gram}_j(k) \geq 0 \text{ and } 1 \leq j \leq N \text{ and } 1 \leq k \leq M \quad (4)$$

$$\text{Len\_Bi\_gram}(k.j) = \prod_{i=1}^{\text{count of (j)}} \text{Len\_Bi\_gram}_{i.i+1}(k.j)$$

As can be seen, i indicates the letters counter, the weight of *j*th word, including *K*th, was computed through bi-gram tagging. In each M-sentence input, there are *N*-word items. In Equation (4), the product of multiplying the occurrence value for each letter with the next letter continues as long as the total number of letters in a word.

E) Position of each word according to the number of sentence words. Member_Word: This matrix indicates the probability of occurrence for each word in a particular position according to its length. First, a matrix trainer is extracted in 194 sentences. In each N-word sentence, each role is observed in a small percentage of cases. Then, the Member_Word matrix of that sentence is obtained according to the length of the input sentence and Equation (5).

$$Member(k.j) \geq 0\ and\ 1 \leq k \leq M \quad (5)$$

$$Member\_Word(k) = \prod_{i=1}^{count\ of\ (j)} Member_{i.i+1}(k)$$

Equation (5) delivers the membership rate of the *k*th sentence from the M-sentence, i indicates the words counter in sentence, including the length of each *N*-length sentence. This matrix is a Bi-gram tagging type.

3. 3. Order of Steps Each training matrix is implemented individually on the input data tested. Once the computations are completed, the success percentage values from each learning matrix, i.e. degree of importance for each training matrix, is obtained. Then, this arrangement is used in the fuzzy GRU steps.

Table 2 provides the success rate for each matrix through a classified fuzzy method from 72 random nonstandard sentences sorted in descending order. In the fuzzy computational section, the sequence of steps for the new fuzzy method can be found in Table 2. Figure 2 compares the success rates of matrices in two standard and nonstandard sentences. In Table 2, the highest success rate with the Bi_gram_combine matrix and with less than 2 % difference, the Transfer _ Bi _ gram matrix is in the second place.

TABLE 2. Order of fuzzy computation classes

No.	Matrix	Success percentage with nonstandard input sentences
1	Bi_gram_combine	66.77%
2	Transfer_Bi_gram	65.14%
3	Transfer_Uni_gram	63.84%
4	Len_Bi_gram_Word	58.95%
5	Member_Word	57.65%

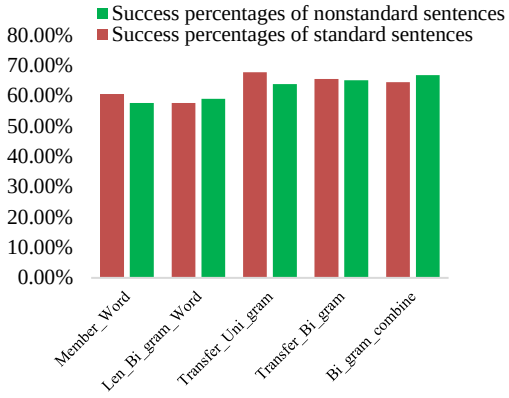

Figure 2. Comparison of success rate in standard and nonstandard input sentences [31]

According to the results in Table 2, the order of 5 steps for the newly proposed computational method is obtained as shown in Figure 2. This arrangement of standard sentences is different from that of nonstandard sentences.

3. 4. Fuzzy GRU Computations The proposed method has been implemented using Visual Studio 2012. Furthermore, the statistics are calculated using Excel 2013 and the results are converted to SQL server 2008. The steps of the proposed method come with a simple format of fuzzy GRU computations illustrated in Figure 3.

The components of Figure 3 are as follows:

C(t-1): The main input in these computations is the output values of the previous steps, where the current step is actually the remaining states of the previous step. In the first step, the value of this input is a matrix to the number of all possible states as well as the array of corresponding values.

C(t): The output of each array step is as long as 20% of the remaining state of the fuzzy computations, which obtained the highest values from the current step computations, as well as the array of possible states values with the same length.

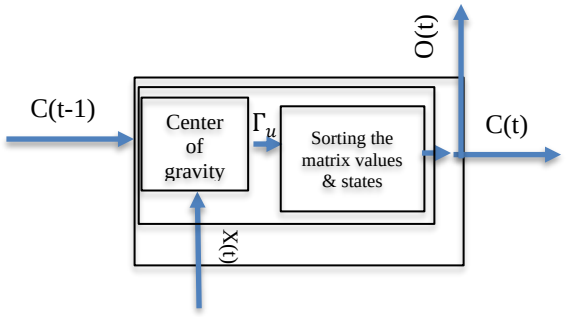

Figure 3. Overall steps of the proposed fuzzy GRU method

X(t): is the values of matrices training in each class, the user's input sentences, and the decomposition of the input sentences.

O(t): indicates removing bottom 80% of possible sentence formulation scenarios as well as reducing 80% of the array length of possible values and array of possible states.

Γ_u: updates gate varies in each class. It updates the array table for sentence states through the center of gravity method after receiving 20% of the highest values of the previous step, the matrix values of the current class and the input sentence states, user's input sentences, as well as the decomposition of input sentences.

According to Table 2, the fuzzy GRU algorithm will be similar to the proposed algorithm. The remarkable point in the new algorithm is that the values are arranged from largest to smallest at each step after computing the roles of possible scenarios. In this procedure, 20% of cases with the highest values are passed to the next step, while discarding the rest.

Proposed algorithm-Classified fuzzy proposed method

1. Obtain the Transfer_Uni_gra, Transfer_Bi_gram,
2. Obtain Bi_gram_Combine, Member_Word, and Len_Word matrices according to the input sentence and the word input composition.
3. Obtain the number of possible states for the input sentence and screening it according to the pair of roles and the composition inserted into the system.
4. Start the first step by entering all possible states as C(t-1) of the first step.
5. The first step is to enter Bi_gram_Combine matrices and user input sentences and decompose input sentences as X(t).
6. The first step of operation Γ_u as the first step of obtaining the center of gravity for all the remaining states with the Bi_gram_Combine matrix.

$$\text{center of gravity } Bi\_gram\_Combine = \frac{\sum_{i=1}^{1-\text{word number}} \text{Bi\_gram\_combine}(the\ role\ of\ \text{i}-1\ th\ word, role\ of\ ith\ word) \times \text{placeof i th word})}{\sum_{i=1}^{1-\text{word number}} \text{place of i th word}} \quad (6)$$

In Equation (6), i indicates the words counter in sentence and indicates how the center of gravity for each input sentence is computed using the membership matrix in the fuzzy decision-making process.

7. The second step of operation Γ_u in the first step: Descending order of possible states and values of possible states.
8. Separate 20% of the largest possible sentence formulation states with maximum values, send those states to the output as C(t) and remove other possible states as O(t) of the current step.
9. Start the second step by inserting C(t-1) as the output of the first step.
10. The second step is to enter the Transfer_Bi_gram matrix and user input sentences and decompose input sentences as X(t).

11. The second step of operation Γ_u in the second step: Obtain the center of gravity for all input states with the Transfer_Bi_gram matrix.

The center of gravity for the matrix values is employed according to the input sentence, the sentence input type and the position of words in each sentence in each of the remaining cases through Equation (7).

$$\text{center of gravity } transfer\_Bi\_gram = \frac{\sum_{i=1}^{1-\text{word number}} (Bi\_gram\_combine(\text{role of i+1th word, type of i th word}) \times \text{placeof i th word})}{\sum_{i=1}^{1-\text{word number}} \text{place of i th word}} \quad (7)$$

Note: Since Bi_gram tagging has been used in Equation (7), the counter changes from the first word to the last remaining word.

12. The second step of operation Γ_u in the second step: Descending order of possible states and values of possible states.

13. Separate 20% of the largest possible sentence formulation states with maximum values, send those states to the output as C(t) and remove other possible states as O(t) of the second step.

14. Start the third step by inserting C(t-1) as the output of the second step.

15. The second step is to enter the Transfer_Uni_gram matrix and user input sentences and decompose input sentences as X(t).

16. The second step of operation Γ_u in the third step: Obtain the center of gravity for all remaining states with the Transfer_Uni_gram matrix according to Equation 8.

$$\text{Center of gravity } Transfer\_\text{uni\_gram} = \frac{\sum_{i=1}^{\text{word number}} \left(Uni\_\text{gram\_combine}(\text{role i th word and type of i th word}) \times \text{place of i th word}\right)}{\sum_{i=1}^{\text{number of words}} \text{place of i th word}} \quad (8)$$

In Equation (8), i indicates the words counter in sentence and indicates how the center of gravity is computed for each input sentence using the Uni_gram_Combine matrix in the fuzzy decision-making process.

17. The second step of operation Γ_u in the third step: Descending order of possible states and values of possible states.

18. Separate 20% of the largest possible sentence formulation states with maximum values, send those states to the output as C(t) and remove other possible states as O(t) of the third step.

19. Start the fourth step by inserting C(t-1) as the output of the second step.

20. The fourth step is to enter the Len_Word matrix and user input sentences and decompose input sentences as X(t).

21. The second step of operation Γ_u in the fourth step: Obtain the center of gravity for all remaining states with the Len_Word matrix.

$$\text{center of gravity } Len\_Word = \frac{\sum_{i=1}^{\text{word number}} (\text{Length of i th word} \times \text{place of ith word})}{\sum_{i=1}^{\text{word number}} \text{place of ith word}} \quad (9)$$

In Equation (9), i indicates the words counter in sentence and displays how to calculate the center of gravity for the word weights according to the resulting Len_Bi_gram and the center of gravity defuzzifier. Descending order of possible states and values of possible states.

22. The second step of operation Γ_u in the fourth step: Descending order of possible states and values of possible states.

23. Separate 20% of the largest possible sentence formulation states with maximum values, send those states to the output as C(t) and remove other possible states as O(t) of the fourth step.

24. Start the fifth (last) step by inserting C(t-1) as the output of the fourth step.

25. The fifth step is to enter Member _Word matrix and user input sentences and decompose input sentences as X(t).

26. The second step of operation Γ_u in the fifth step: Obtain the center of gravity for all remaining states with the Member_Word matrix.

$$\text{center of gravity } member = \frac{\sum_{i=1}^{\text{word number}} (\text{membership of i th word} \times \text{place of i th word})}{\sum_{i=1}^{\text{word number}} \text{place of i th word}} \quad (10)$$

In Equation (10), i indicates the words counter in sentence and indicates how the center of gravity for each remaining sentence is computed using the membership matrix in the fuzzy decision-making process.

27. The second step of operation Γ_u in the fifth step: Descending order of possible states and values of possible states.

28. Obtain the largest value of the remaining values and the corresponding state. Send that state to the output as the roles of that input sentence.

As can be seen in the newly proposed algorithm, the computational load in the new method in each step was curtailed by 80% compared to its previous class. These matrices contain fuzzy values requiring no fuzzification tools. In contract, matrices in fuzzy morphological techniques (e.g. [17]) are first fuzzified with complex sequences. In this respect, the newly proposed method offers an optimal computational load.

Therefore, the quasi-code of the newly proposed algorithm can be considered as follows: Algorithm 1- Recurrent Neural Networks with Gated Recurrent Unit

```
1.  Function Sort_As(Matrix_Method by val
    ,MatrixASe_InDe by ref, MatrixSe_InDe by ref,
    MatrixASe_De by ref, MatrixSe_De by ref)
2.  {
3.  MatrixASe_InDe =Make Matrix_Method
    with Pos_word && MatrixSe_InDe;
4.  MatrixASe_De =Make Matrix_Method with
    Pos_word && MatrixSe_De;
5.  Sort Ascending MatrixASe_InDe &&
    MatrixSe_InDe , MatrixASe_De &&
    MatrixSe_De ;
6.  Get Twenty percent of the largest numbers:
```

```
    MatrixASe_InDe && MatrixSe_InDe,
    MatrixASe_De && MatrixSe_De;
7.  Return MatrixASe_InDe && MatrixSe_InDe,
    MatrixASe_De && MatrixSe_De;
8.  }
9.  Const Matrixes:
    Transfer_Uni_gram،Transfer_Bi_gram ،
    Bi_gram_Combine ،Member_Word ,Len_Word;
10. Input Se: Get sentences from the user;
11. Input ASe: Get analyses for Sentences from the user;
12. Begin
13. For  i=1 to Len(Sentences)
14. Set  MatrixSe_InDe (15^Len(SE[i]))
    &&MatrixSe_De (9^Len(SE[i])) as string;
    //Matrix for status of Independent and
    dependent Roles
15. Set MatrixASe_InDe(15^Len(SE[i])) &&
    MatrixASe_De(9^Len(SE[i])) as single;
    //Matrix for Value_status of Independent
    and dependent Roles
17. Get All Value in Matrix: MatrixSe_InDe,
    MatrixASe_InDe, MatrixSe_De,
    MatrixASe_De;
18. Call Sort_As (COG_Bigram_combine,
    MatrixASe_InDe , MatrixSe_InDe,
    MatrixASe_De , MatrixSe_De) //eq6-Step1
19  Call Sort_As (COG_Transfer_Bigram,
    MatrixASe_InDe , MatrixSe_InDe,
    MatrixASe_De , MatrixSe_De) //eq7-step2
20  Call Sort_As (COG_Transfer_Uni_gram,
    MatrixASe_InDe , MatrixSe_InDe,
    MatrixASe_De ,
    MatrixSe_De) //eq8-step3
21. Call Sort_As (COG_Len_Word ,
    MatrixASe_InDe , MatrixSe_InDe,
    MatrixASe_De , MatrixSe_De) //eq9-step4
22. Call Sort_As (COG_Member_Word,
    MatrixASe_InDe , MatrixSe_InDe,
    MatrixASe_De , MatrixSe_De) //eq10-step5
23. Next i
24. Output : MatrixSe_InDe(1) , MatrixSe_De(1);
25. End
```

4. RESULTS

In addition to the roles provided in Section 3, there are "verb-letter" roles, which were discarded because of their shared decomposition and composition.

$$\text{Success percentages} = \frac{\text{Success} \times 100}{Total} \quad (11)$$

Relying on Equation (11), we obtained the success rate in each section. The *total number* in Equation (11) changes in each section. In addition, the *success rate* in this regard varies according to each section.

4. 1. Overall Success Rate in Persian Sentences Composition The test results in this method indicates that the fuzzy classified overall success rate in nonstandard sentences is 68.73% in all roles. In standard sentences, however, the success rate is 67% in identification of all roles. In this case, the new fuzzy method outperformed by roughly 2%.

Therefore, in 72 nonstandard sentences inserted into the system to test the newly proposed method, 211 out of 307 roles are correctly detected. In Equation (11), 307 is inserted for Total Number, while 211 is inserted for Success Rate, which together deliver Figure 4.

4. 2. Success Percentage in Independent Roles Among 307 roles in 72 sentences, 152 of them were related to independent roles. Of the 152 independent roles in nonstandard sentences, the tests suggested that 110 were correctly detected, with a success rate of 72.37% based on Equation (11) (Figure 4). In the standard sentences, however, the success rate was approximately 70% in 106 cases. Figure 5 illustrates the values.

If we want to examine the roles separately and calculate the success rates (according to Equation (11)), Table 3 is obtained.

As can be seen, the highest success rate in nonstandard input sentences is *predicate*, whereas this role assumes the lowest value in standard sentences.

Figure 6 compares the success rates of two sentence structures in the independent role category. As for *subject*, the success values for the two role categories

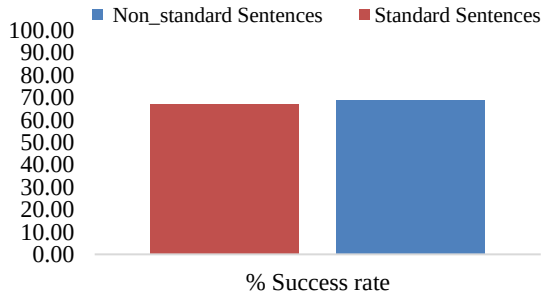

Figure 4. The success rates of standard and nonstandard sentences in general [26]

Figure 5. The success rates of independent and dependent roles in two types of standard input [26] and nonstandard sentences

TABLE 3. Success rates for independent roles of standard and nonstandard input sentences [26]

Independent roles	Success percentages of independent roles in nonstandard sentences	Success percentages of independent roles in standards sentences
Predicate	75.76	60.61
Subject	72.53	73.63
Object	66.67	61.11
Complement	70	80.00

were roughly equal. Only in *complement*, the success rates of standard sentences are higher than those of nonstandard sentences.

4. 3. Success Percentage in Dependent Roles

Among 307 roles in 72 sentences, 155 of them were related to dependent roles. Of 155 dependent roles in nonstandard sentences, 101 cases were correctly obtained. According to Equation (11), success rate was 65.16% for nonstandard sentences. This value is 64.52% for standard sentences, which has been compared in Figure 5.

If we want to examine the roles separately, Table 4 is obtained based on Equation (11).

Some words in Persian sentences may not take any dependent roles. Alternatively, the new method may not identify any roles for certain words, thus sending *unknown* to output.

Table (4) displays the success rate for each of the 12 dependent roles in both standard and nonstandard Persian sentence inputs. Rows 1 to 6 indicate frequently used dependent roles in Persian sentences, while rows 7 to 12 cover roles less commonly found in Persian sentences. As shown in Table (4), the highest success values in most frequently used roles were achieved in nonstandard sentences. In less commonly used roles except for *bending*, however, the success values were equal.

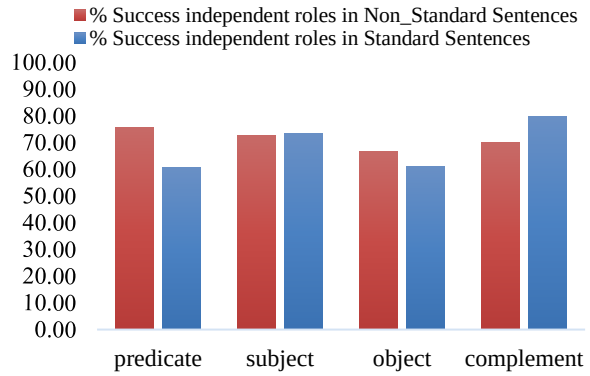

Figure 6. The success rate of each independent role in two types of standard [26] and nonstandard sentences

TABLE 4. The success rates of dependent roles in both standard [26] and nonstandard sentences

No.	Independent and dependent roles	Success percentages of independent roles in nonstandard sentences	Success percentages of independent roles in standards sentences
1	Adjective	33.33	55.56
2	Noun	54.55	72.73
3	Adverb	96.55	89.66
4	N.A.	57.14	61.22
5	Governing genitive	76.92	46.15
6	Genitive	76.92	46.15
7	Apposition	100	100
8	Governing transducer	100	100
9	Bending	25	75
10	Retroactive	100	75
11	Exclamation	50	50
12	Annunciator	50	50

Figure 7 compares the success rates of dependent roles in standard and nonstandard Persian sentences. The noteworthy point in this figure is that the success rate decreases because of low presence of the rare roles in Persian sentences (marked in blue/proposed method). For that reason, success rates were 100% and 50% in apposition-governing transducer and exclamation-annunciator, respectively. Another point is that most dependent roles appear in pairs in Persian sentences [31].

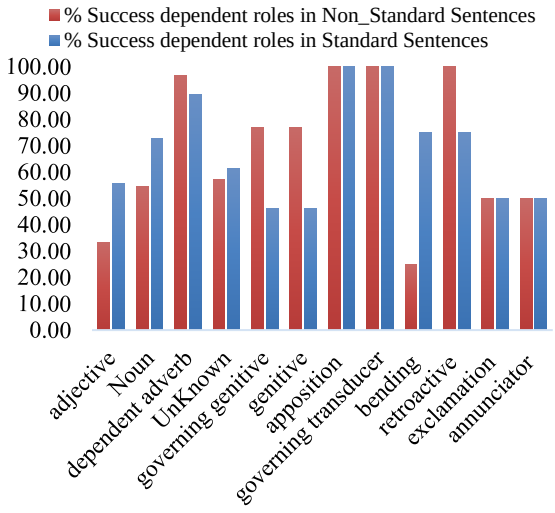

Figure 7. The success rate of each dependent role in two types of standard [26] and nonstandard sentences

5. CONCLUSION AND FUTURE WORK

Considering that the current study covers nonstandard sentences of Persian language, the results obtained by the newly proposed method demonstrate how it provides better results in sentences that barely follow Persian language grammar. Hence, the greater success rate in obtaining sentence roles was achieved by standard [26] and nonstandard sentences separately for two categories of independent and dependent roles. Independent roles are better when comparing the success rates of two categories of nonstandard sentence roles. In all independent roles except complement, the success of the proposed method is achieved with nonstandard sentences. Similarly, in all dependent roles except noun, adjective, noun and bending, the proposed method has performed better on nonstandard sentences than standard sentences.

In this paper, the memory of the new fuzzy GRU method allows the researcher to select irregular values relative to the input states. This is because 20% of the highest values in the current step may include the first, middle, last, or *N*th values of the input states (output of the previous step). On the other hand, the fuzzy nature of the newly proposed method allows us to use the fuzzy values of the matrices obtained using statistical methods to train the possible states desirably.

Reducing the computation of steps in different steps of this algorithm is one of the most important advantages compared to [8]. When employing the fuzzy method [8], if the amount of computations for each class is 100,000, the total will be 500,000, while it is exactly 124960 in the fuzzy GRU. In fact, there is nearly as much as $\frac{4}{5}$ reduction in computations compared to [8]. Therefore, further investigation and research in the field of neural network computing, deep fuzzy recurrent learning can provide Persian-Arabic language linguists with clear results for this topic from the data mining progress. In this regard, one of the areas to be covered in future research is that, the sequence of steps 2-3 in each of the independent and dependent roles can be obtained separately and the results can be discussed.

Moreover, the research will adopt the second fuzzy type in order to take advantage of the uncertainty of this method and ultimately enhance the results.

6. REFERENCES

1. "Persian language, Encyclopædia Britannica.", Written by The Editors of Encyclopaedia Britannica, Retrieved from https://www.britannica.com/topic/Persian-language
2. Alshammari, M., Nasraoui, O., and Sanders, S. "Mining Semantic Knowledge Graphs to Add Explainability to Black Box Recommender Systems." ***IEEE Access***, Vol. 7, (2019), 110563–110579. https://doi.org/10.1109/access.2019.2934633
3. Heo, Y., Kang, S., and Yoo, D. "Multimodal Neural Machine Translation with Weakly Labeled Images." ***IEEE Access***, Vol. 7, (2019), 54042–54053. https://doi.org/10.1109/ACCESS.2019.2911656
4. Wu, B., Li, K., Ge, F., Huang, Z., Yang, M., Siniscalchi, S. M., and Lee, C. H. L. "An end-to-end deep learning approach to simultaneous speech dereverberation and acoustic modeling for robust speech recognition." ***IEEE Journal on Selected Topics in Signal Processing***, Vol. 11, No. 8, (2017), 1289–1300. https://doi.org/10.1109/JSTSP.2017.2756439
5. Vani, H., and Anusuya, M. "Fuzzy Speech Recognition: A Review." ***International Journal of Computer Applications***, Vol. 177, No. 47, (2020), 39–54. https://doi.org/10.5120/ijca2020919989
6. Keysers, D., Deselaers, T., Rowley, H. A., Wang, L. L., and Carbune, V. "Multi-Language Online Handwriting Recognition." ***IEEE Transactions on Pattern Analysis and Machine Intelligence***, Vol. 39, No. 6, (2017), 1180–1194. https://doi.org/10.1109/TPAMI.2016.2572693
7. Jayashree, R., Murthy, S. K., and Sunny, K. "Keyword extraction based summarization of categorized Kannada text documents." ***International Journal on Soft Computing***, Vol. 2, No. 4, (2011), 81–93. https://doi.org/10.5121/ijsc.2011.2408
8. Motameni, H., and Peykar, A. "Morphology of compounds as standard words in Persian through hidden Markov model and fuzzy method." ***Journal of Intelligent and Fuzzy Systems***, Vol. 30, No. 3, (2016), 1567–1580. https://doi.org/10.3233/IFS-151865
9. Graves, A. "Generating Sequences With Recurrent Neural Networks." arXiv:1308.0850, Vol. 5, (2014), 1–43. Retrieved from http://arxiv.org/abs/1308.0850
10. Dim Lam, C., and Khin Mar, S. Joint Word Segmentation and Part-of-Speech Tagging for Myanmar Language, PhD Dissertation, University of Computer Studies, Yangon. Retrieved from http://onlineresource.ucsy.edu.mm/handle/123456789/2530
11. Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." arXiv:1412.3555, (2014). Retrieved from http://arxiv.org/abs/1412.3555
12. Obin, N., and Lanchantin, P. "Symbolic modeling of prosody: From linguistics to statistics." ***IEEE Transactions on Audio, Speech and Language Processing***, Vol. 23, No. 3, (2015), 588–599. https://doi.org/10.1109/TASLP.2014.2387389
13. Shin, Y., and Xu, C. Intelligent systems: modeling, optimization, and control. CRC press, 2017.
14. Natel Khanlari, P. "Linguistics and Persian Language." Tehran: Toos Publication, 1987.
15. Li, F. K. "A Short History of Linguistics, R. H. Robins." ***American Anthropologist***, Vol. 70, No. 6, (1968), 1186–1186. https://doi.org/10.1525/aa.1968.70.6.02a00210
16. Moniri, M. "Fuzzy and intuitionistic fuzzy turing machines." ***Fundamenta Informaticae***, Vol. 123, No. 3, (2013), 305–315. https://doi.org/10.3233/FI-2013-812
17. Taheri, A., Meghdari, A., Alemi, M., and Pouretemad, H. R. "Teaching music to children with autism: A social robotics challenge." ***Scientia Iranica***, Vol. 26, No. 1, (2019), 40–58. https://doi.org/10.24200/sci.2017.4608
18. Mitkov, R. The Oxford handbook of computational linguistics, Oxford University Press, 2004.
19. Tatar, D. "Word Sense Disambiguation by Machine Learning Approach: A Short Survey." ***Fundamenta Informaticae***, Vol. 64, No. 1–4, (2005), 433–442.
20. Hinrichs, E. W., Meurers, W. D., and Wintner, S. "Linguistic Theory and Grammar Implementation: Introduction to this Special Issue." ***Research on Language and Computation***, Vol. 2,

No. 2, (2004), 155–163. https://doi.org/10.1023/b:rolc.0000016748.09606.a9

21. Bijankhan, M., Sheykhzadegan, J., Bahrani, M., and Ghayoomi, M. "Lessons from building a Persian written corpus: Peykare." ***Language Resources and Evaluation***, Vol. 45, No. 2, (2011), 143–164. https://doi.org/10.1007/s10579-010-9132-x
22. Shamsfard, M., Ilbeygi, M., and Sadat Jafari, H. "STeP-1: A Set of Fundamental Tools for Persian Text Processing." In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), (2010), 859–865. Retrieved from https://www.researchgate.net/publication/220746093
23. Assi, S. M. "Grammatical Tagging of a Persian Corpus." ***International Journal of Corpus Linguistics***, Vol. 5, No. 1, (2000), 69–81. https://doi.org/10.1075/ijcl.5.1.05ass
24. Natural Language Processing Software of Ferdowsi University of Mashhad Version 1.3.(persian)," Web Technology Lab of Ferdowsi University of Mashhad, Mashhad, 2012.
25. Safari, A., Hosseini, R., and Mazinani, M. "A Novel Type-2 Adaptive Neuro Fuzzy Inference System Classifier for Modelling Uncertainty in Prediction of Air Pollution Disaster." ***International Journal of Engineering, Transactions B: Applications***, Vol. 30, No. 11, (2017), 1746–1751. https://doi.org/10.5829/ije.2017.30.11b.16
26. Sadeghi, H., Motameni, H., Ebrahimnejad, A., and Vahidi, J. "Morphology of composition functions in Persian sentences through a newly proposed classified fuzzy method and center of gravity defuzzification method." ***Journal of Intelligent and Fuzzy Systems***, Vol. 36, No. 6, (2019), 5463–5473. https://doi.org/10.3233/JIFS-181330
27. Haspelmath, M., and Sims, A. Understanding morphology, London: Hodder Education and Hachette UK Company, 2010.
28. Geeraerts, D., and Cuyckens, H. The Oxford Handbook of Cognitive Linguistics. Oxford University Press, 2007. https://doi.org/10.1093/oxfordhb/9780199738632.001.0001
29. Perry, J. R. "Persian Morphology." In Morphologies of Asia and Africa, Winona, EIS - Eisenbrauns, (pp. 975–1019), 2007.
30. Amidi, A., Amidi, S., 'Super VIP Cheatsheet: Machine Learning,' https://stanford.edu/~shervine/, stanford, 2018.
31. Peykar, A. Pars Process Persian sentence analyzer software, Gorgan: Golestan University, Faculty of Basic Sciences, 2011.

Persian Abstract

چکیده

سرعت روبه‌رشد ورود املای کلمات مختصر و حضور کلمات خارجی و شکلک‌ها در زبان فارسی اهمیت پژوهش‌های داده‌کاوی در این زبان را دوچندان می‌سازد، از طرفی تشابه املایی–دستوری زبان فارسی به عربی نشان دهنده‌ی آن است که می‌توان در سایر زبان‌های مشابه نیز از این پژوهش استفاد کرد. در این راستا این پژوهش به داده‌کاوی جملات غیر استاندارد زبان فارسی در جهت یافتن نقش هر کلمه در جمله می‌پردازد. میزان محاسبات با روش‌های سنتی در هر کدام از پنج عامل پذیرش نقش حداقل تعداد محاسبات ((۵×تعدادکلمات ۹)+ (۵×تعدادکلمات ۱۵)) است که ممکن است خارج از توان روش‌های سنتی پردازش زبان‌های طبیعی می‌باشد، بنابراین در این پژوهش از روش GRU برای پردازش این محاسبات استفاده شده است. روش پیشنهادی حاضر با استفاده از دو دسته نقش‌های "مستقل- وابسته" و پنج عامل پذیرش نقش کلمات در جملات به عنوان پنج دروازه ساخت خروجی، نتایج شناسایی نقش کلمات را تقویت می‌بخشد. مقادیر جداول آموزش دهنده‌ی این روش، فازی هستند؛ بنابراین برای تصمیم‌گیری درباره مقادیر فازی و نیز کاهش پیچیدگی و ابهام این محاسبات، از روش فازی مرکزثقل استفاده شده است. به طور خلاصه می‌توان این روش پیشنهادی را "GRU فازی" نامید. نتایج نشان می دهد که روش پیشنهادی، کاهش ۸۰٪ میزان محاسبات در هر دروازه به روزسانی و تقویت تقریبا ۲٪ از ٦٧٪ در جملات استاندارد به ٦٩٪ جملات غیر استاندارد را دارا است.