Digital Poetry Production Using Word Embedding Technology

Ali Abdali [1,*]

[1] Control Systems and Instrumentation Department of Iranian Offshore Oil Company, Kharg, Bushehr 7546143572, Iran
* Corresponding Author

Received: 19/10/2024, Revised: 08/11/2024, Accepted: 20/11/2024.

Abstract

Natural language processing or NLP is a branch of artificial intelligence that deals with the interaction between computers and human language. The goal of NLP is to give machines the ability to understand, analyze, and generate natural language. Natural language processing is an interdisciplinary science that tries to facilitate the interaction between human language and computer by using computer science, linguistics and artificial intelligence. In this paper, we present a method for producing synthetic digital poetry using word embedding technology. Word embedding is a technique used to represent words as numerical vectors in multidimensional space. This technique allows NLP to better understand the semantic and syntactic connections between words. The importance and advantages of using the word vector are discussed below. Machine learning models using word vectors have higher accuracy in analysis and prediction. These models can learn more complex relationships and patterns.

Keywords

Digital Poetry, Word Embedding, Neural Networks.

1. Introduction

Digital poetry is developing day by day as a new attitude and as a new artistic genre in line with the advancement of technology. In this type of poetry, the poet becomes a programmer who produces poetry by combining machine language and human language. This genre has grown significantly in recent years along with the issues raised in the field of computational linguistics. Computational linguistics is one of the interdisciplinary sciences in which the machine modeling of natural language is done using mathematical methods. With the processing of natural language, the presence of artificial intelligence in digital poetry becomes more visible and a new discourse begins in the text. In the programming of this unconventional type of poetry, several algorithms are used in the text game [1,2]. The word embedding is one of the fundamental elements of natural language processing that helps to better understand language and improve the performance of NLP systems. With the advancement of technology and the development of new models, the importance and applications of word vector continue to expand. The technology of text generation using artificial intelligence dates back to the early years of the 1950s. During this period, basic research in the field of artificial intelligence and natural language processing began [3-5]. Early systems such as ELIZA (developed in the 1960s) attempted to generate human text using simple rules. At this time, most attention was focused on linguistic analysis and generating simple sentences. In the 1990s, with the advancement of machine learning algorithms, statistical models such as n-grams and Hidden Markov Models (HMMs) were developed. These models are better able to simulate natural language. In the years after 2010, the emergence of neural networks, especially recurrent neural networks (RNNs) and later transformer networks, created a great transformation in text generation. Models such as GPT (Generative Pre-trained Transformer) and BERT (Bidirectional Encoder Representations from Transformers) emerged, which dramatically increased the quality of text production. These models can be used in various fields such as creative writing, translation, summarizing and answering questions. This technology is developing rapidly and a bright future is predicted in the field of text production and its applications in various industries. Using optimization algorithms in the text production process can help improve the quality and efficiency of text production. These algorithms allow us to produce texts with more desirable properties, such as coherence, diversity, and accuracy. Genetic algorithms (GA) are one of the optimization methods inspired by the natural processes of natural selection and evolution. In this method, a set of generated texts can be considered as a population of solutions and then gradually optimized. This algorithm is used in the production of digital poetry [6-8].

Simulated Annealing is another text generation algorithm. This algorithm is inspired by the process of gradual cooling of materials. This method can be used to search the answer space and generate optimal texts. In this method, starting from an initial text and making small changes such as changing words or sentences are used to produce larger texts. Also, in some researches, machine learning technique has been used to produce Bangla Poetry [9]. In the articles [10-12], methods about thematic poems and how to analyze them are examined. In different languages, due to the difference in the structure of poetry, the way of processing digital poetry is different, among others, we can mention research in this field in Chinese language, which investigates a solution to produce poetry in Chinese language using neural networks. Pay [13]. Also, in some researches, strategies for producing Bahasa Indonesia poetry are examined [14]. Among other works done in this field and in the Japanese language, artificial intelligence methods in the production of Haiku can be mentioned [15]. Also, in the field of producing meaningful poetic texts using genetic algorithm, there have been researches and researches, most of the efforts are to arrange sentences and transfer a semantic unit from the poetic text to the audience [16]. Also, in the field of computational models and ways of thinking in prose and poetry, there have been significant studies and researches in which the structure of language and grammar in machine text is closer to the text written by a human author [17], [18]. Optimization using deep learning is also one of the important methods in text production and digital poetry production. Deep learning models such as Recurrent Neural Networks (RNN) or Transformers can be trained using optimization methods such as Adam or SGD to produce higher quality texts. Deep learning models such as LSTM or GRU can be used to generate poetry. These models can be trained using input word vectors and generating subsequent words. Using recurrent neural networks (RNN) is one of the main techniques in text generation. Due to their special structure, RNN networks are able to process data sequences such as text and have been very successful in generating natural language. Unlike conventional neural networks, RNNs have the ability to consider information from previous sequences and depend on each step in the input chain. This feature makes them very useful for problems such as natural language processing (NLP), because the order of words in a sentence is very important [19-23].

In this paper, we will use word vector and deep learning to generate synthetic digital poetry. In the first part, we will explain the word embedding technology and its important role in the process of producing digital text and poetry. Then, in the second part, we describe its famous models, and by converting words into numerical vectors, we process the input data of neural networks, and in the third part, we describe our method for producing digital poetry using word vectors. In the fourth part, we examine the simulation of the model based on neural networks, in the fifth part, we examine the challenges related to semantic understanding. At the end and in the conclusion section, we present the limitations, obstacles and future ways of poetry production.

2. Word Embedding

Word Embedding is one of the common methods in natural language processing (NLP) that represents words as numerical vectors in multidimensional space. The purpose of using word vectors is to display the concepts and meanings of words in such a way that they can be used in machine learning algorithms and neural networks. In general, word vectors are constructed by models such as Word2Vec, GloVe, and FastText. By learning from a huge volume of texts, these models assign a vector to each word that represents the semantic features of that word. Vectors belonging to similar words will be close to each other in the vector space.

The important features of the word vector can be mentioned as follows:

• Low dimensionality, rich meaning: Each word is represented as a multidimensional vector (eg 300 dimensions). Each dimension encodes different information about the meaning and usage of the word.

• Distance and semantic similarity: Euclidean or cosine distance between the vectors of two words can indicate their semantic similarity.

• Semantic composability: using algebraic operations (such as addition and subtraction of vectors) it is possible to obtain approximately the meaning of new expressions or combinations. For example:

Gender calculation:

Sister ≈ man - brother + woman In this example, if we have the vector of the word "man" and subtract the vector of "brother" from it, the concept of "dominion" is somehow obtained regardless of gender. Then by adding the vector "woman", we get the word "sister". Figure 1

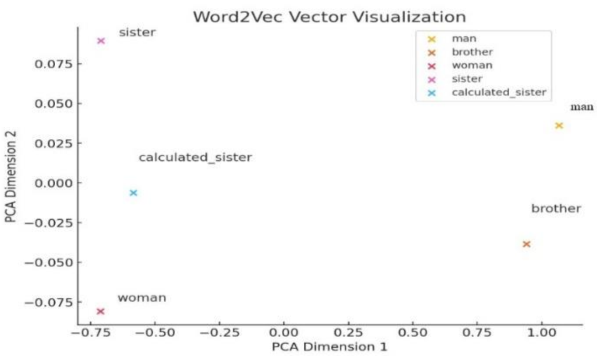

Fig.1 Two-dimensional diagram of the word2vec model based on word meaning categories.

Calculation of nationality:

Moscow - Russia + Turkey ≈ Ankara

This operation states that "Moscow" has the same relation to "Russia" as "Ankara" has to "Turkey". By subtracting the vector "Russia" from "Moscow" and adding the vector "Turkey", we can arrive at "Ankara". Figure 2

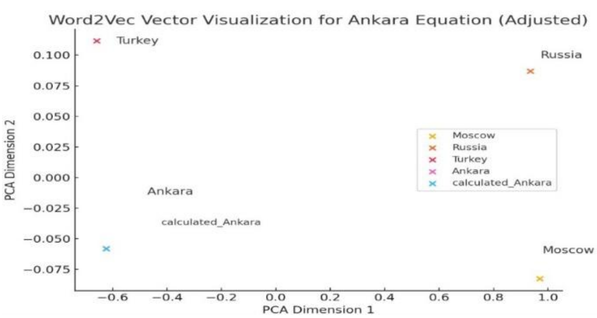

Fig.2 Two-dimensional diagram of the word2vec model based on word meaning categories.

Job relationships and roles:

Doctor - hospital + university ≈ professor

DOI: 10.22034/cmes.2024.2043770.1035

- In this example, if "doctor" is related to "hospital", by replacing "hospital" with "university", we can get a similar role, "professor".

Change seasons:

Summer - heat + cold ≈ winter

- The vectors related to the characteristics of the seasons, such as "heat" for summer and "cold" for winter, can be well combined in this operation.

Tools and jobs:

Painter + brush ≈ carpenter + hammer

- This relationship somehow shows the relationship between jobs and the tools used in them. For example, "painter" has the same relation to "brush" as "carpenter" has to "hammer".

Kinship:

father + mother ≈ parents

- The sum of the two vectors "father" and "mother" usually results in a vector that represents the meaning of "parents" well.

These operations are derived using word vector models such as Word2Vec and GloVe and show how semantic relations can be achieved through vector computation.

3. Training models to produce digital poetry

We used the following two models in the production of digital poetry in this paper and trained word vector models using large textual data in Persian literature. Training models in the process of producing poetry is one of the most important steps that allows the model to learn language patterns, meanings and relationships between words and then produce meaningful and logical poems. In this section, the process of training linguistic models, characteristics of input data and different methods of training in the production of digital poetry are discussed.

• Word2Vec:

Skip-Gram model: In this model, the goal is to predict the surrounding words (surrounding words) using a target word. In other words, for each word, the model tries to predict which words are near it.

Continuous Bag of Words (CBOW) model: This model does the opposite. That is, it uses the surrounding words to predict the target word.

By training these models, each word is assigned to a numerical vector that shows the semantic connections between words in multidimensional space.

• GloVe (Global Vectors for Word Representation):

Instead of predicting words from each other, this method works based on the co-occurrence matrix of words. GloVe tries to use this matrix to create vectors that represent the meaning of words in relation to their co-occurrences.

3.1 The process of training models in the production of digital poetry

In this paper, we used the Word2Vec model, which is one of the most important and popular methods for representing words as numerical vectors to produce digital poetry. This model is widely used in natural language processing (NLP). Word2Vec is trained in two general modes:

1. Continuous Bag of Words (CBOW): The central word prediction model according to neighboring words.

2. Skip-gram: a model for predicting neighboring words according to the central word

3.1.1 Word2Vec training steps

3.1.1.1 Data collection (input text)

First, a text dataset is needed, which may be collected from various sources such as books, articles, news, etc. This text should contain a large amount of real language data so that the model can learn the semantic relationships between words well.

3.1.1.2 Pre-processing of the poem

Before starting to train the model, the poetic text must be pre-processed. In this method, we applied the following pre-processing to the input data to produce digital poetry in Persian, which includes the following:

• Removal of punctuation marks: such as periods, commas and question and exclamation marks.

• Convert to lowercase: all words are converted to lowercase.

• Stop Words: Words such as "and", "or", "of" which usually have no special meaning are removed.

• Tokenization: breaking text into tokens (words).

3.1.1.3 Choosing the word window (Window Size)

In Word2Vec, every word in a sentence is related to the words around it. The window size determines how many words before and after the target word the model will pay attention to. For example, if the window size is 2, the model will only pay attention to two words before and two words after the target word.

3.1.1.4 Model selection: CBOW or Skip-gram

Word2Vec is trained in two ways:

Continuous Bag of Words (CBOW):

In this method, the model tries to predict the central word (target) according to the neighboring words. The model first combines the neighboring word vectors and then a neural network is used to predict the central word. This method is more suitable for smaller data sets and when speed is important. In this paper, we used this method to produce digital poetry according to our case study, which was a small data set. Word2Vec uses a simple two-layer neural network (one input layer and one output layer) to learn word vectors. The learning process is such that each word in the input layer is converted into a vector (usually initialized with random numbers).

Using a cost function (such as Negative Sampling) or Hierarchical Softmax), the weights are updated such that words have similar vectors that are close to each other.

3.1.1.5 Optimization and learning

During training, the model uses methods such as backpropagation and Stochastic Gradient Descent (SGD) to improve the weights of the neural network. This process is repeated until the model optimally learns the relationships between words.

3.1.1.6 Model output:

After completing the training, each word in the dataset was converted into a numerical vector. These vectors are displayed as points in a multidimensional space. Word

DOI: 10.22034/cmes.2024.2043770.1035

vectors that are semantically close to each other have a small distance from each other in this space.

Two stanzas from a sample poem produced using the CBOW model

A woman is standing by the wall

And the beauty that has come down a few steps from the sky

In the first stanza, the CBOW model uses neighboring words ("woman", "beside", "wall", "is") to predict "standing".

The training of the neural network in this study was conducted using the Word2Vec model, specifically the Continuous Bag of Words (CBOW) method. This approach leverages a dataset of Persian literary texts to create word vectors that capture semantic relationships between words. During training, each word in the input text is represented as a numerical vector, which the network learns to optimize through techniques like backpropagation and stochastic gradient descent (SGD). The primary goal of this training is to enable the model to generate coherent, meaningful, and contextually accurate lines of poetry by predicting target words based on their surrounding context. Ultimately, the objective is to create a system that can autonomously generate poetic text that resonates with the structural and emotional nuances characteristic of human-written poetry.

4. Network structure

The CBOW model is a neural network that has an input layer and an output layer, which is used to learn the word vector. We taught:

• Input: neighboring words enter the network as one-hot vectors.

In the CBOW model, the input consists of neighboring words. Suppose for a target word t whose neighboring words C1, C2, ..., Cn, we have:

Input Vector=one-hot(C1) +one-hot(C2) +…+one-hot (Cn)

• Hidden layer: vectors of neighboring words are averaged and converted into a combined vector.

If W is the weight matrix of the hidden layer with dimensions N×V (where V is the total number of words in the dictionary and N is the dimension of the word vector), then to calculate the hidden vector h we have:

h= $1/n \sum_{i=1}^{n} Wci$

where Wci represents the neighboring word vector ci.

• Output: The model predicts what the central word (target word) is using the Softmax layer.

We use the hidden vector h to predict the target word t. The output vector U is calculated as follows:

u=W′h

where W ′ is the weight matrix of the output layer with dimensions V × N.

In this network, we used the cost function (Negative Sampling).

And through gradient descent and backpropagation, the weights were updated, which made the model gradually improve.

4.1 Mathematical Equations in the CBOW Model

Let's assume the following:

- Wt
- C: the number of words in the context window.
- $Wt - c$, $Wt-(c-1)$, …, $Wt-1$, $Wt+1$, …, $Wt+c$: the context words .
-
-

4.2 Averaging the context word vectors

First, each context word (from the window) is mapped to its embedding vector. Let Vw be the embedding vector of word w. Then, the embedding vectors of the context words are summed:

$$\mathrm{V_{context}} = 1/2\mathrm{C} \sum_{i=-C,\, i\neq 0}^{C} Vw_{t+i}$$

This means the average of the context word vectors is computed. This forms the input to the CBOW model.

4.3 Predicting the target word

The model uses the vector $\mathrm{V_{context}}$ to predict the probability of the target word Wt

This is done using the softmax function:

$$\mathrm{P}(\, Wt \mid Wt-c, \dots, Wt+c\,) = \frac{\exp\,(V_{Wt}^{T}\ \mathrm{V_{context}})}{\sum_{w\in V} \exp\,(V_{W}^{T}\ \mathrm{V_{context}})}$$

where:

- V: the set of all words in the vocabulary.
- Wt : the embedding vector of word w.

4.4 Loss Function

The loss function used to optimize the model is typically the cross-entropy. For each pair of target and context words, the loss is defined as:

L= $-\ \log \mathrm{P}(\, Wt \mid Wt-c, \dots, Wt+c\,)$

This loss is computed and used to update the model parameters (i.e., the word embedding vectors) through optimization methods such as stochastic gradient descent (SGD).

5. Challenges related to semantic understanding

Poetry production by linguistic models and text production algorithms faces specific and more complex challenges compared to normal text production. Due to the unique nature of poetry, these challenges include special language structures, rhythm, emotions, and complex artistic images that require a high level of semantic and artistic understanding to understand and produce.

5.1 Metaphors and similes

One of the main challenges in producing poetry is understanding and producing metaphors and similes. Metaphors and similes are one of the main poetic tools to express feelings and complex concepts indirectly. Linguistic models are often unable to understand or generate complex metaphors because these concepts are usually dependent on human mental and emotional experiences.

For example, in the Persian language, understanding the metaphor "the sea is heartbroken" requires understanding the symbolic connection between "the sea" and "feeling heartbroken", which goes beyond the normal linguistic patterns and requires a deeper interpretation of the meaning.

5.2 Ambiguity and Multilayered Meaning

One of the prominent features of poetry is its multi-layeredness and semantic ambiguity. In poetry, many words and sentences may be interpreted in several ways. For example, a word like "fire" in Persian literature can both refer to its real meaning (i.e. flame) and metaphorically refer to love, anger, or inner transformation. Text generation models are usually based on statistical data and ordinal patterns, and they can hardly cope with the understanding and generation of multi-layered meaning in poetry. These models may assume a superficial or incorrect meaning for words and may not have a correct understanding of complex metaphors and symbolism.

5.3 Maintaining rhythm

Poetry usually follows a rhythmic pattern that contributes to the musical flow and beauty of the text. Rhymes and weights are very important in many classical poems (such as Persian ghazal or English) and producing a text that respects these rules is one of the main challenges. Combining rhythm and rhyme with the right meaning is a problem that is challenging even for humans. Linguistic models may be unable to produce correct rhymes while preserving the exact meaning of the text, especially in languages with complex rhyming and weight structures. In many cases, models may produce texts that are semantically correct but not rhyme or rhythm, or vice versa.

5.4 Maintaining sentiment and tone

One of the most important aspects of poetry is conveying emotions and appropriate tone. Poetry is often a means of expressing human emotions such as love, sadness, happiness, and anger. Therefore, not only the meaning of the words must be correct, but the emotions behind them must also be correctly conveyed. Text generation models are usually unable to accurately detect the emotions and tone of the lyrics. They may not be able to correctly discern when the tone should be formal, romantic, or emotional. For example, a model designed to generate romantic poetry might produce sentences that convey inappropriate or out-of-context sentiments.

5.5 Semantic Coherence and Thematic Consistency

A good poem should usually have thematic coherence and semantic coherence. Poems, despite being short, often convey a single message or feeling completely. Text generation models may have difficulty maintaining thematic coherence, especially over longer texts. Many models cannot properly maintain the semantic connection between different lines of the poem. The result may be a text in which each line is acceptable independently, but which as a whole does not convey a clear meaning or message. For example, one line might be about love and the next about nature, with no meaningful connection between them.

5.6 Polysemy

Poetry often uses ambiguity as an artistic device. The poet may deliberately use words that have multiple meanings to invite the reader to different interpretations. Text generation models often have difficulty choosing the appropriate meaning for a word in poetic contexts. Models may choose one of the meanings of a polysemous word, while the poet's intended meaning may be more complex and multi-layered.

5.7 Creativity and Innovation

Poetry is inherently a creative activity that depends on innovation in language and meaning. Poets often combine words in new ways, use unusual structures, and go beyond common language patterns. Linguistic models are highly dependent on their training data and are therefore limited in innovation and creation of new linguistic combinations. These models usually do not have the ability to create something completely new and creative and are more concerned with reproducing known linguistic structures.

6. Conclusion

Producing poetry through linguistic models is more challenging than producing prose due to its semantic, emotional and structural complexity. Understanding and producing multi-layered meanings, complex metaphors, rhythm and maintaining thematic continuity are among the main problems. To improve this process, we need models that can access deeper cultural, historical, and semantic knowledge in addition to linguistic patterns and can correctly reproduce the creative aspects of language. In this paper, we tried to produce poetry by artificial intelligence by presenting a language model for digital poetry production in Persian language. Although the poems produced by this model as given in the examples of this article are acceptable and thinkable, artificial intelligence still faces challenges compared to human creativity in artistic creation, most of these challenges are rooted in hardware limitations. It seems that in the near future, by increasing the processing power of microprocessors based on artificial intelligence, it will be possible to process a higher volume of data, and there is a possibility of reducing the challenges presented in this article about artificial poems. In such a way that the aesthetics of poetry based on artificial intelligence and the creativity provided by it may overcome the poetic productions of great human poets. In any case, it doesn't matter whether the poem has a human origin or a machine origin, and what is important is that the poem is something that happens in the language. Future research in digital poetry generation should focus on enhancing semantic depth, emotional resonance, rhythmic precision, and creative language use, ultimately aiming to bridge the gap between artificial and human poetic expression.

References

[1] Ali Abdali, Digital poetry production using neural networks, Computational methods in engineering sciences, vol. 1, no. 4, Winter 2024

[2] P. Kumar, S. Manikandan, R. Kishore, A Novel Approach for Text Generation using RNN for Language Modeling, 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)

[3] Nourizadeh H, Nazar MS. Customer-oriented scheduling of active distribution system considering integrated demand response programs and multi-carrier

energy hubs. Journal of Cleaner Production. 2024 Apr 1;447:141308.

[4] Nourizadeh H. Long-term Planning of Optimal Placement of Distribution Transformers to Improve Reliability and Power Quality with the Approach of Reducing Costs and Losses. Iranian Electric Industry Journal of Quality and Productivity. 2020 Mar 10;9(1):49-58.

[5] Nourizadeh H, Niasati M. Optimal Placement of Distribution substations with The Purpose of Reducing Losses and Energy Not Supplied by A Multi-Objective Function and Artificial Bee Colony (ABC) Optimization Algorithm. Computational Intelligence in Electrical Engineering. 2020 Dec 21;11(4):65-80.

[6] Ali. Abdali, Designing the Artificial Poet Using Genetic Algorithm, Kingston University, London, http://epc.buffalo.edu/e-poetry/2013/ ExtendedProgramme.pdf, June 17-20, 2013

[7] Prakash, Kolla Bhanu, Sanjeevikumar Padmanaban, and Massimo Mitolo. Smart and Power Grid Systems–Design Challenges and Paradigms. River Publishers, 2023.

[8] Padmanaban, Sanjeevikumar, et al. "Energy Storage." Smart and Power Grid Systems–Design Challenges and Paradigms. River Publishers, 2023. 211-236.

[9] Diaz-Agudo, Belen, Pablo Gervas, and Pedro A Gonzalez-Calero Poetry generation in COLIBRI. European Conference on Case-Based Reasoning springer Berlin Heidelberg, 2002.

[10] Erica Greene, Tugba Bodrumlu, and Kevin Knight. 2010. Automatic analysis of rhythmic poetry with Applications to generation and translation. In Proc. EMNLP.

[11] Hisar Manurung. An evolutionary algorithm approach to poetry generation. PhD. thesis, University of Edinburgh. 2003.

[12] Zhe Wangy, Wei Hez, Hua Wuz, Haiyang Wuz, Wei Liz, Haifeng Wangz, Enhong Cheny , Chinese Poetry Generation with Planning based Neural Network, arXiv:1610.09889v2 [cs.CL] 7 Dec 2016.

[13] Fam Rashel, Ruli Manurung Poetry Generation for Bahasa Indonesia using a Constraint Satisfaction Approach, ISBN: 978-979-1421-19-5 ICACSIS 2013.

[14] Miroslava Hroncová, Kristína Machová, Generating Haiku Poems Using Method of Artificial Intelligence, ISBN 978-80-553-2271-1 WIKT 2015.

[15] Ruli Manurung a, Graeme Ritchie b, Henry Thompson, using genetic algorithms to create meaningful poetic text, Journal of Experimental & Theoretical Artificial Intelligence Vol. 24, No. 1, March 2012, 43–64.

[16] Sarjoun Doumit, Nagendra Marupaka and Ali A Minai, “Thinking in Prose and Poetry: A Semantic Neural Model” IJCNN, IEEE, 4 - 9 Aug, 2013.

[17] David M. Kaplan, David M. Blei “A Computational Approach to Style in American Poetry, Seventh IEEE International Conference on Data Mining, 1550-4786/07 25.00 © 2007 IEEE.

[18] S. Joshua Johnson, M. Ramakrishna Murty, I. Navakanth, A detailed review on word embedding techniques with emphasis on word2vec, Springer Published: 03 October 2023, Volume 83, pages 37979–38007, (2024)

[19] Hongzhong Wang, Kun Guo, Zhanghui Liu, Mixed Word Embedding Method Based on Knowledge Graph Augment for Text Classification, 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

[20] Rifat Rahman, Robust and Consistent Estimation of Word Embedding for Bangla Language by fine-tuning Word2Vec Model, 2020 - 23rd International Conference on Computer and Information Technology (ICCIT)

[21] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, Furu Wei, Improving Text Embeddings with Large Language Models, arXiv:2401.00368v3 [cs.CL] ,(2024)

[22] Ali Abdali, Verya Monjezi, Machine Learning-based Flexible Link Robot Control, Computational methods in engineering sciences, (2023)

[23] Deepak Suresh Asudani, Naresh Kumar Nagwani, Pradeep Singh, Impact of word embedding models on text analytics in deep learning environment, Springer, Volume 56, pages 10345–10425, (2023)