Linguistic Constructions Translation Method Based on Neural Networks

Eugene Fedorov and Olga Nechyporenko

Cherkasy State Technological University, Shevchenko blvd., 460, Cherkasy, 18006, Ukraine

Abstract

The paper proposes a linguistic constructions translation method based on recurrent neural networks. The novelty of the study lies in the fact that to ensure the interaction of software agents representing subjects within supply chains, four artificial neural network models for the translation of the linguistic structures were created, a criterion for evaluating the training effectiveness of the proposed models was selected, and the parameters of the proposed models were identified based on the Adam method. In the created models, unlike the existing translational neural networks, the decoder does not have feedback from the output layer to the hidden layer. The developed models and methods for their parametric identification make it possible to improve the accuracy of translation of natural language constructions. The created natural language constructions translation method based on neural networks can be used in various intelligent computer systems that use the translation of linguistic constructions.

Keywords

supply chain, multi-agent interaction, artificial neural network, translations of linguistic constructions, Adam method, linguistic constructions

1. Introduction

Currently, one of the important problems in the field of natural language processing is the insufficiently high probability, adequacy, and speed of translation [1, 2]. This leads to the fact that the natural language interaction of computer agents in multi-agent systems may be inefficient. Thus, the creation of models and methods that increase the efficiency of using natural language constructs for the interaction of computer agents is an urgent task.

The work aims to develop a natural language constructions translation method. To achieve the goal, the following tasks were set and solved:

- analyze existing methods of translation;
- propose neural network models of translation;
- propose a criterion for evaluating the effectiveness of neural network translation models;
- create methods for determining the values of parameters of neural network translation models;
- perform a numerical study.

2. Literature review

In this paper, the interaction of computer agents of multi-agent systems representing subjects that interact within supply chains is chosen as the scope of linguistic constructions [3-5].

Examples of multi-agent systems for supply chains are [3, 4]:

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine
EMAIL: fedorovee75@ukr.net (E. Fedorov); olne@ukr.net (O. Nechyporenko)
ORCID: 0000-0003-3841-7373 (E. Fedorov); 0000-0002-3954-3796 (O. Nechyporenko)

– Agent Building Shell. Coordinating the actions of agents representing the firm and supply chain actors interacting with it (for example, suppliers and customers) using the COOL coordination language;

– MetaMorph. Coordination of the actions of agents representing the firm, supply chain actors interacting with it and intermediaries, using the actions of intermediaries;

– NetMan. Management of agents representing business units of firms and interacting within the same firm and between firms through agreements;

– BPMAT &SCL. BPMAT models firm activity, SCL models inter-firm flows;

– MASCOT. Coordinating the actions of agents representing the company and the subjects of the supply chain interacting with it. Used to improve supply chain flexibility through planning and scheduling. Coordinates production across multiple sites and evaluates new products and strategic business decisions (such as production, purchase, or supplier selection) based on capacity and material needs across the entire supply chain);

– DASCh. Management of agents representing the firm, product flows, and information flows to investigate gaps in these flows;

– Task dependency network. Management of agents representing the firm and supply chain entities interacting with it using the auction protocol (selection of agents through an auction);

– MASC. Management of agents representing the firm and supply chain entities interacting with it using the auction protocol (selection of agents through an auction);

– OCEAN. Management of agents representing the firm and the subjects of the supply chain interacting with it through negotiations. Uses competition at the local level and cooperation at the global level.

Specified multi-agent systems do not provide computer simulations of the interaction of supply chain entities based on linguistic constructs and soft computing.

Today, artificial intelligence methods are used to translate linguistic constructions, while the most popular is the connectionist approach [6-8], which for many networks allows the use of parallel learning methods [9-11].

The following recurrent networks are most often used as neural networks for translation:

- Elman neural network (ENN or SRN) [12, 13], the simplest of recurrent neural networks;
- bidirectional recurrent neural network (BRNN) [14, 15], which is built based on two Elman neural networks;
- long short-term memory (LSTM) [16, 17];
- bidirectional recurrent neural network (BLSTM) [18, 19], which is built based on two LSTM neural networks;
- gated recurrent unit (GRU) [20, 21];
- bidirectional recurrent neural network (BGRU) [22], which is built based on two GRU neural networks.

The advantages of neural networks are [20, 23]:

- the possibility of their training and adaptation;
- the ability to identify patterns in the data, their generalization, i.e., extraction of knowledge from data, therefore knowledge about the object (for example, its mathematical model) is not required;
- parallel processing of information, which increases computing power.

The disadvantages of neural networks are [24, 25]:

- the high probability of the learning and adaptation method hitting a local extremum;
- the difficulty of determining the structure of the network, since there are no algorithms for calculating the number of layers and neurons in each layer for specific applications;
- inaccessibility for human understanding of the knowledge accumulated by the network (it is impossible to represent the relationship between input and output in the form of rules), since they are distributed among all elements of the neural network and are presented in the form of its weight coefficients;
- the difficulty of forming a representative sample.

Thus, none of the networks satisfies all the criteria.

3. Proposed methodology

3.1. Modified neural network Seq2seq

Figure 1 shows proposed in this article a modified Seq2seq neural network for translation, which is a recurrent network. The modified Seq2seq neural network includes an encoder and a decoder (unlike the classical Seq2seq neural network [26, 27], there is no feedback from the output layer to the hidden layer in the decoder).

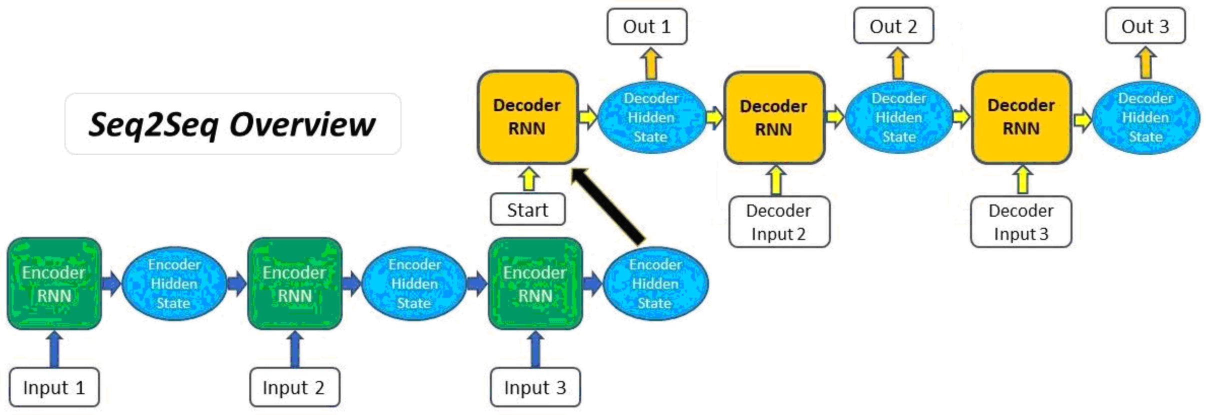

Figure 1: Proposed modified neural network Seq2seq

Let us limit our consideration to the functioning of the ANN and the case when the encoder and decoder are based on ENN.

It is assumed that a text input sequence of length P has already been converted to an encoded input sequence $\mathbf{x}$ (for example, by word2vec), and the length of each encoded word of the input sequence is $N^{(0)}$. It is assumed that an encoded output sequence $\mathbf{y}$ of length P will be converted to a text output sequence, and the length of each encoded word of the output sequence is equal to $N^{(3)}$. The number of hidden layers in the encoder and decoder is equal to $N^{(1)}$ and $N^{(2)}$ respectively.

Functioning of the ANN

1. Initialization

$$n=1,\ m=1,\ \mu=1,$$
$$y^{(1)}_{n-1,i}=0,\ i\in\overline{1,N^{(0)}},$$
$$y^{(2)}_{m-1,i}=0,\ i\in\overline{1,N^{(2)}}.$$

2. Calculation of the output signal for the hidden layer of the encoder

$$y^{(0)}_{ni}=x_{\mu i},\ i\in\overline{1,N^{(0)}},$$
$$y^{(1)}_{nj}=f^{(1)}(s^{(1)}_{nj}),\ j\in\overline{1,N^{(1)}},$$
$$s^{(1)}_{nj}=b^{(1)}_{j}+\sum_{i=1}^{N^{(0)}}w^{(1)}_{ij}y^{(0)}_{ni}+\sum_{i=N^{(0)}+1}^{N^{(0)}+N^{(1)}}w^{(1)}_{ij}y^{(1)}_{n-1,i-N^{(0)}}.$$

3. Checking the completion of encryption. If $n<P$, then $\mu=\mu+1$, $n=n+1$, go to 2.
4. Calculation of the output signal for the hidden and output layers of the decoder

$$y^{(2)}_{mj}=f^{(2)}(s^{(2)}_{mj}),\ j\in\overline{1,N^{(2)}},$$

$$s_{mj}^{(2)} = b_j^{(2)} + \sum_{i=1}^{N^{(1)}} w_{ij}^{(2)} y_{Pi}^{(1)} + \sum_{i=N^{(1)}+1}^{N^{(1)}+N^{(2)}} w_{ij}^{(2)} y_{m-1,i-N^{(1)}}^{(2)} ,$$

$$y_{mj} = y_{mj}^{(3)} = f^{(3)}(s_{mj}^{(3)}), \ j \in \overline{1, N^{(3)}} ,$$

$$s_{mj}^{(3)} = b_j^{(3)} + \sum_{i=1}^{N^{(2)}} w_{ij}^{(3)} y_{mi}^{(2)} .$$

5. Checking the completion of the decryption. If $m < P$, then $m = m+1$, go to 4.

3.2 Modified additive attention neural network

Figure 2 shows proposed in this article a modified additive attention neural network for translation (proposed by Bahdanau), which is a recurrent network. The modified additive attention neural network includes an encoder, an attention mechanism, and a decoder (unlike the classical additive attention neural network [28], there is no feedback from the output layer to the hidden layer in the decoder). The attention mechanism allows the decoder to focus attention on certain outputs of the encoder.

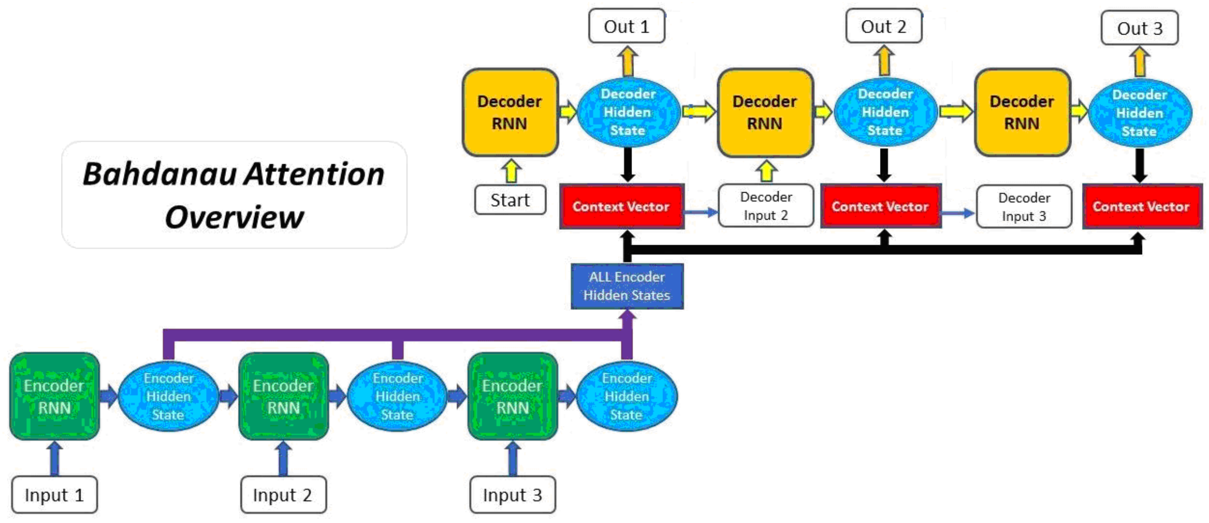

Figure 2: Proposed modified additive attention neural network

Let us limit our consideration to the functioning of the ANN and the case when the encoder and decoder are based on ENN.

It is assumed that a text input sequence of length *P* has already been converted to an encoded input sequence $\mathbf{x}$ (for example, by word2vec), and the length of each encoded word of the input sequence is $N^{(0)}$. It is assumed that an encoded output sequence $\mathbf{y}$ of length *P* will be converted to a text output sequence, and the length of each encoded word of the output sequence is equal to $N^{(3)}$. The number of hidden layers in the encoder and decoder is equal to $N^{(1)}$ and $N^{(2)}$ respectively.

Functioning of the ANN

1. Initialization

$$n=1, \ m=1, \ \mu=1,$$

$$y_{n-1,i}^{(1)} = 0, \ i \in \overline{1, N^{(0)}} ,$$

$$y_{m-1,i}^{(2)} = 0, \ i \in \overline{1, N^{(2)}} .$$

2. Calculation of the output signal for the hidden layer of the encoder

$$y_{ni}^{(0)} = x_{\mu i},\ i \in \overline{1, N^{(0)}},$$

$$y_{nj}^{(1)} = f^{(1)}(s_{nj}^{(1)}),\ j \in \overline{1, N^{(1)}},$$

$$s_{nj}^{(1)} = b_j^{(1)} + \sum_{i=1}^{N^{(0)}} w_{ij}^{(1)} y_{ni}^{(0)} + \sum_{i=N^{(0)}+1}^{N^{(0)}+N^{(1)}} w_{ij}^{(1)} y_{n-1,i-N^{(0)}}^{(1)}.$$

3. Checking the completion of encryption. If $n < P$, then $\mu = \mu + 1$, $n = n + 1$, go to 2.

4. Additive attention. Concatenative (additive) attention is used, which connects the encoder's hidden layer and the decoder's hidden layer.

4.1. Calculation of weights (estimates) of attention

$$e_{mn} = \sum_{j=1}^{N^{(1)}} \left(w3_j \tanh\left(\sum_{i=1}^{N^{(1)}} w1_{ij} y_{ni}^{(1)} + \sum_{i=1}^{N^{(2)}} w2_{ij} y_{m-1,i}^{(2)} \right) \right),\ n \in \overline{1, P},$$

$$a_{mn} = \text{softmax}(e_{mn}) = \frac{\exp(e_{mn})}{\sum_{l=1}^{P} \exp(e_{ml})},\ n \in \overline{1, P}.$$

4.2. Context calculation

$$c_{mi} = \sum_{n=1}^{P} a_{mn} y_{ni}^{(1)},\ i \in \overline{1, N^{(1)}}.$$

5. Calculation of the output signal for the hidden and output layers of the decoder

$$y_{mj}^{(2)} = f^{(2)}(s_{mj}^{(2)}),\ j \in \overline{1, N^{(2)}},$$

$$s_{mj}^{(2)} = b_j^{(2)} + \sum_{i=1}^{N^{(1)}} w_{ij}^{(2)} c_{mi} + \sum_{i=N^{(1)}+1}^{N^{(1)}+N^{(2)}} w_{ij}^{(2)} y_{m-1,i-N^{(1)}}^{(2)},$$

$$y_{mj} = y_{mj}^{(3)} = f^{(3)}(s_{mj}^{(3)}),\ j \in \overline{1, N^{(3)}},$$

$$s_{mj}^{(3)} = b_j^{(3)} + \sum_{i=1}^{N^{(2)}} w_{ij}^{(3)} y_{mi}^{(2)}.$$

6. Checking the completion of decryption. If $m < P$, then $m = m + 1$, go to 4.

3.3 Modified multiplicative attention neural network

Figure 3 shows proposed in this article a modified multiplicative attention neural network for translation (proposed by Luong), which is a recurrent network. The modified multiplicative attention neural network includes an encoder, an attention mechanism, and a decoder (unlike the classical multiplicative attention neural network [29], there is no feedback from the output layer to the hidden layer in the decoder). The attention mechanism allows the decoder to focus attention on certain outputs of the encoder.

Let us limit our consideration to the functioning of the ANN and the case when the encoder and decoder are based on ENN.

It is assumed that a text input sequence of length *P* has already been converted to an encoded input sequence $\mathbf{x}$ (for example, by word2vec), and the length of each encoded word of the input sequence is $N^{(0)}$. It is assumed that an encoded output sequence $\mathbf{y}$ of length *P* will be converted to a text output sequence, and the length of each encoded word of the output sequence is equal to $N^{(3)}$. The number of hidden layers in the encoder and decoder is equal to $N^{(1)}$ and $N^{(2)}$ respectively.

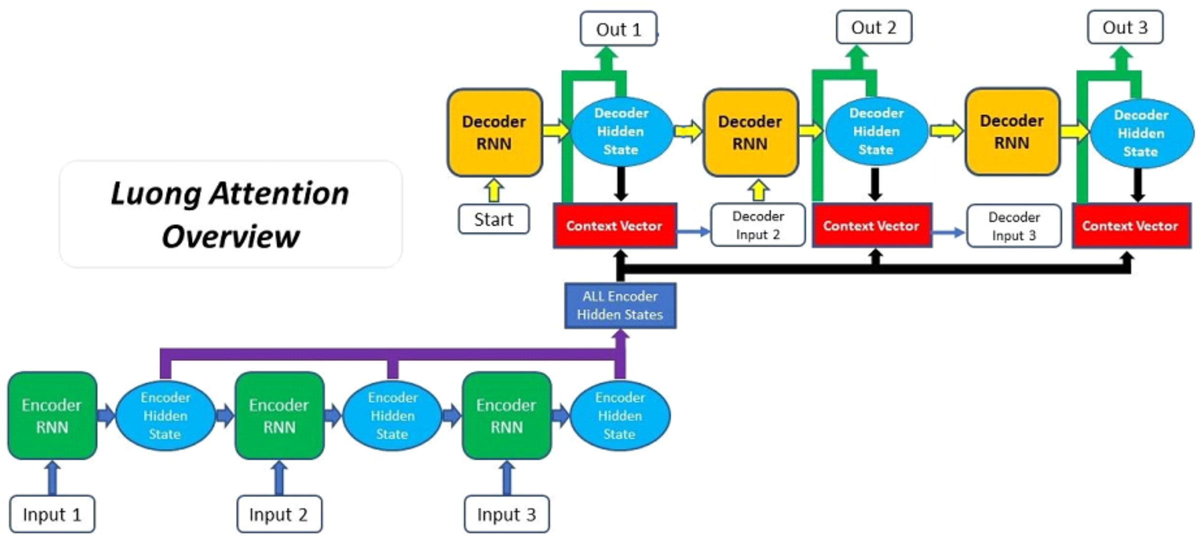

Figure 3: Proposed modified multiplicative attention neural network

Functioning of the ANN

1. Initialization

$$n=1,\ m=1,\ \mu=1,$$

$$y^{(1)}_{n-1,i}=0,\ i\in\overline{1,N^{(0)}},$$

$$y^{(2)}_{m-1,i}=0,\ i\in\overline{1,N^{(2)}}.$$

2. Calculation of the output signal for the hidden layer of the encoder with the concatenation of all outputs

$$y^{(0)}_{ni}=x_{\mu i},\ i\in\overline{1,N^{(0)}},$$

$$y^{(1)}_{nj}=f^{(1)}(s^{(1)}_{nj}),\ j\in\overline{1,N^{(1)}},$$

$$s^{(1)}_{nj}=b^{(1)}_j+\sum_{i=1}^{N^{(0)}}w^{(1)}_{ij}y^{(0)}_{ni}+\sum_{i=N^{(0)}+1}^{N^{(0)}+N^{(1)}}w^{(1)}_{ij}y^{(1)}_{n-1,i-N^{(0)}}.$$

3. Checking the completion of encoding. If $n<P$, then $\mu=\mu+1$, $n=n+1$, go to 2.

4. Calculation of the output signal for the hidden layer of the decoder

$$y^{(2)}_{mj}=f^{(2)}(s^{(2)}_{mj}),\ j\in\overline{1,N^{(2)}},$$

$$s^{(2)}_{mj}=b^{(2)}_j+\sum_{i=1}^{N^{(2)}}w^{(2)}_{ij}y^{(2)}_{m-1,i}.$$

5. Multiplicative attention. Multiplicative attention is used, which links the hidden layer of the encoder and the hidden layer of the decoder.

5.1. Calculation of weights (estimates) of attention:

- if the multiplicative attention is "general", then

$$e_{mn}=\sum_{j=1}^{N^{(1)}}\sum_{i=1}^{N^{(2)}}y^{(2)}_i(m)\,w_{ij}\,y^{(1)}_j(n),\ n\in\overline{1,P};$$

- if the multiplicative attention is "dot" and $N^{(2)}=N^{(1)}$, then

$$e_{mn}=\sum_{i=1}^{N^{(2)}}y^{(2)}_{mi}y^{(1)}_{ni},\ n\in\overline{1,P},$$

$$a_{mn} = \mathrm{softmax}(e_{mn}) = \frac{\exp(e_{mn})}{\sum_{l=1}^{P} \exp(e_{ml})}, \ n \in \overline{1, P}.$$

5.2. Context calculation

$$c_{mi} = \sum_{n=1}^{P} a_{mn} y_{ni}^{(1)}, \ i \in \overline{1, N^{(1)}}.$$

6. Calculation of the output signal for the hidden and output layers of the decoder

$$\widehat{y}_{mj}^{(2)} = \tanh\left(\widehat{b}_j^{(2)} + \sum_{i=1}^{N^{(1)}} \widehat{w}_{ij}^{(2)} c_{mi} + \sum_{i=N^{(1)}+1}^{N^{(1)}+N^{(2)}} \widehat{w}_{ij}^{(2)} y_{m,i-N^{(1)}}^{(2)} \right), \ j \in \overline{1, N^{(2)}},$$

$$y_{mj} = y_{mj}^{(3)} = f^{(3)}(s_{mj}^{(3)}), \ j \in \overline{1, N^{(3)}},$$

$$s_{mj}^{(3)} = b_j^{(3)} + \sum_{i=1}^{N^{(2)}} w_{ij}^{(3)} y_{mi}^{(2)}.$$

7. Checking the completion of decryption. If $m < P$, then $m = m + 1$, go to 4.

3.4 An additive attention neural network with pointing

In this paper, we propose an additive attention neural network with an indication for translation (similar to Figure 2), which is a recurrent network. The author's additive attention neural network with pointing includes an encoder, an attention mechanism, a pointing mechanism, and a decoder (unlike the classical additive attention neural network [30], the decoder does not have feedback from the output layer to the hidden one and the pointing mechanism is added). The attention mechanism allows the decoder to focus attention on specific encoder outputs. The pointing mechanism allows the decoder to focus on specific encoder inputs.

Let us limit our consideration to the functioning of the ANN and the case when the encoder and decoder are based on ENN.

It is assumed that a text input sequence of length *P* has already been converted to an encoded input sequence $\mathbf{x}$ (for example, by word2vec), and the length of each encoded word of the input sequence is $N^{(0)}$. It is assumed that an encoded output sequence $\mathbf{y}$ of length *P* will be converted to a text output sequence, and the length of each encoded word of the output sequence is equal to $N^{(3)}$. The number of hidden layers in the encoder and decoder is equal to $N^{(1)}$ and $N^{(2)}$ respectively.

Functioning of the ANN

1. Initialization

$$n = 1, \ m = 1, \ \mu = 1,$$

$$y_{n-1,i}^{(1)} = 0, \ i \in \overline{1, N^{(0)}},$$

$$y_{m-1,i}^{(2)} = 0, \ i \in \overline{1, N^{(2)}}.$$

2. Calculation of the output signal for the hidden layer of the encoder

$$y_{ni}^{(0)} = x_{\mu i}, \ i \in \overline{1, N^{(0)}},$$

$$y_{nj}^{(1)} = f^{(1)}(s_{nj}^{(1)}), \ j \in \overline{1, N^{(1)}},$$

$$s_{nj}^{(1)} = b_j^{(1)} + \sum_{i=1}^{N^{(0)}} w_{ij}^{(1)} y_{ni}^{(0)} + \sum_{i=N^{(0)}+1}^{N^{(0)}+N^{(1)}} w_{ij}^{(1)} y_{n-1,i-N^{(0)}}^{(1)}.$$

3. Checking the completion of encryption. If $n < P$, then $\mu = \mu + 1$, $n = n + 1$, go to 2.

4. Additive attention. Concatenative (additive) attention is used, which connects the encoder's hidden layer and the decryptor's hidden layer.

4.1. Calculation of weights (estimates) of attention

$$e_{mn} = \sum_{j=1}^{N^{(1)}} \left(w3_j \tanh\left(\sum_{i=1}^{N^{(1)}} w1_{ij} y_{ni}^{(1)} + \sum_{i=1}^{N^{(2)}} w2_{ij} y_{m-1,i}^{(2)} \right) \right), \ n \in \overline{1,P},$$

$$a_{mn} = \operatorname{softmax}(e_{mn}) = \frac{\exp(e_{mn})}{\sum_{l=1}^{P} \exp(e_{ml})}, \ n \in \overline{1,P}.$$

4.2. Context calculation

$$c_{mi} = \sum_{n=1}^{P} a_{mn} y_{ni}^{(1)}, \ i \in \overline{1, N^{(1)}}.$$

5. Pointing

$$y_{mj}^{(3)} = sigm(s_{mj}^{(3)}), \ j \in \overline{1, N^{(2)}},$$

$$s_{mj}^{(3)} = b_j^{(3)} + \sum_{i=1}^{N^{(0)}} w_{ij}^{(3)} y_{ni}^{(0)} + \sum_{i=1}^{N^{(0)}+N^{(1)}} w_{ij}^{(3)} c_{m,i-N^{(0)}} + \sum_{i=N^{(1)}+1}^{N^{(0)}+N^{(1)}+N^{(2)}} w_{ij}^{(3)} y_{m-1,i-N^{(0)}-N^{(1)}}^{(2)}.$$

6. Calculation of the output signal for the hidden and output layers of the decoder

$$y_{mj}^{(2)} = f^{(2)}(s_{mj}^{(2)}), \ j \in \overline{1, N^{(2)}},$$

$$s_{mj}^{(2)} = b_j^{(2)} + \sum_{i=1}^{N^{(1)}} w_{ij}^{(2)} c_{mi} + \sum_{i=N^{(1)}+1}^{N^{(1)}+N^{(2)}} w_{ij}^{(2)} y_{m-1,i-N^{(1)}}^{(2)},$$

$$y_{mj} = y_{mj}^{(4)} = y_{mj}^{(3)} f^{(4)}(s_{mj}^{(4)}) + (1 - y_{mj}^{(3)}) \sum_{n=1}^{P} a_{mn}, \ j \in \overline{1, N^{(3)}},$$

$$s_{mj}^{(4)} = b_j^{(4)} + \sum_{i=1}^{N^{(2)}} w_{ij}^{(4)} y_{mi}^{(2)}.$$

7. Checking the completion of decryption. If $m < P$, then $m = m + 1$, go to 4.

3.5 Criteria for evaluating the effectiveness of neural network translation models

In this work, for the training of neural networks translation models, the criterion of model adequacy was chosen, which means the choice of such values of parameters W that delivers maximum accuracy (the coincidence of the model output and the desired output):

$$F = \frac{1}{P} \sum_{\mu=1}^{P} [\mathbf{d}_\mu = \mathbf{y}_\mu] \to \max_W, \ [\mathbf{d}_\mu = \mathbf{y}_\mu] = \begin{cases} 1, & \mathbf{d}_\mu = \mathbf{y}_\mu \\ 0, & \mathbf{d}_\mu \neq \mathbf{y}_\mu \end{cases}. \tag{1}$$

Training of neural network translation models is subject to criterion (1).

3.6 Method for determining the values of parameters of neural network translation models based on the Adam method

Let the weight vector be defined as

$$\mathbf{w}(n) = \left(w_{11}^{(1)}(n), \ldots, w_{N^{(L-1)} N^{(L)}}^{(L)}(n) \right)^T = \left(w_1(n), \ldots, w_{N_w}(n) \right)^T,$$

where L – maximum number of layers.

Let the ANN energy error be defined as

$$E(n) = \frac{1}{2} \sum_j e_j^2(n), \ e_j(n) = d_j(n) - y_j(n),$$

where $y_j(n)$ – output of the jth neuron of the output layer,

$d_j(n)$ – training output of the jth neuron of the output layer.

Let the vector of partial derivatives (gradient) be defined as

$$\mathbf{g}(n) = \left(\frac{\partial E(n)}{\partial w_1(n)}, \ldots, \frac{\partial E(n)}{\partial w_{N_w}(n)} \right)^T .$$

Step 1. Initialization.

Step 1.1. The initial vector of the weights $\mathbf{w}(0)$ is set.

Step 1.2. The initial vector of the first moments $\mathbf{m}(-1) = \mathbf{0}$ is set.

Step 1.3. The initial vector of the second moments $\mathbf{v}(-1) = \mathbf{0}$ is set.

Step 1.4. Parameter η is set, which determines the learning rate (typically $\eta = 0.001$), decay rates of the first and second moments are set as β_1 and β_2 respectively, $\beta_1, \beta_2 \in [0,1)$ (typically $\beta_1 = 0.9$ and $\beta_2 = 0.999$), and a stability parameter of ε is set to prevent division by zero (typically $\varepsilon = 10^{-8}$).

Step 1.5. An initial gradient of $\mathbf{g}(0)$ is calculated.

Step 1.6. n=0.

Step 2. The vector of the first moments is calculated based on the exponential moving average

$$\mathbf{m}(n) = \beta_1 \mathbf{m}(n-1) + (1-\beta_1)\mathbf{g}(n) .$$

Step 3. The second moment vector is calculated based on the exponential moving average

$$\mathbf{v}(n) = \beta_2 \mathbf{v}(n-1) + (1-\beta_2)\mathbf{g}^2(n) .$$

Step 4. The vector of weights is calculated (the moments are corrected due to their initialization by zero and the training step is scaled):

- traditional variant

$$\hat{\mathbf{m}}(n) = \mathbf{m}(n) / \left(1-\beta_1^{n+1}\right), \; \hat{\mathbf{v}}(n) = \mathbf{v}(n) / \left(1-\beta_2^{n+1}\right),$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\eta \hat{\mathbf{m}}(n)}{\sqrt{\hat{\mathbf{v}}(n)} + \varepsilon};$$

- variant AMSGrad

$$\hat{\mathbf{v}}(n) = \max\{\hat{\mathbf{v}}(n-1), \mathbf{v}(n)\}, \text{ where } \hat{\mathbf{v}}(-1) = \mathbf{0},$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\eta \mathbf{m}(n)}{\sqrt{\hat{\mathbf{v}}(n)} + \varepsilon}.$$

Step 5. Gradient $\mathbf{g}(n+1)$ is calculated.

4. Experiments and results

The numerical study of the proposed methods for determining the parameter values was carried out in the Google Colaboratory environment using the Tensorflow package.

To determine the structure of the Seq2seq modified neural network model, additive attention, multiplicative attention, additive attention with pointing with 256 input neurons, i.e., determining the number of hidden neurons, several experiments were carried out, the results of which are presented in Figure 4.

The standard data set spa-eng (English-Spanish dictionary) was used as input data to determine the values of the parameters of the neural network translation model from http://www.manythings.org/anki. The data set contained 20000 records (18000 for training and 2000 for validation). The criterion for choosing the structure of the neural network model was translation accuracy.

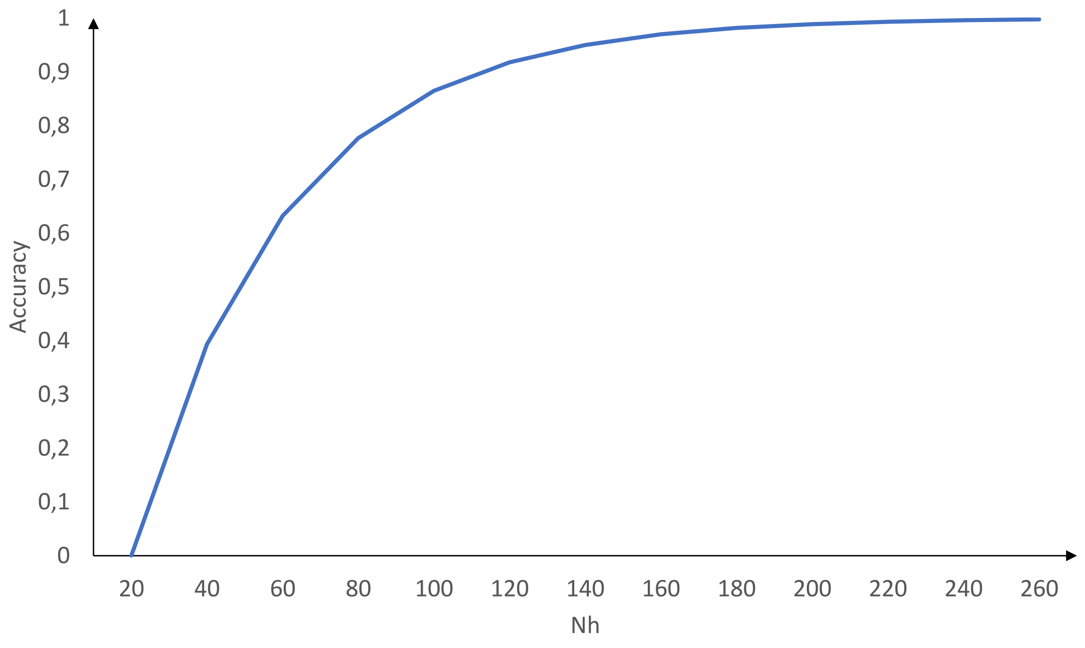

Figure 4: Graph of the dependence of the translation accuracy value on the number of hidden neurons

As can be seen from Figure 4, with an increase in the number of hidden neurons, the accuracy value increases. For translation, it is sufficient to use 256 hidden neurons (corresponding to the number of input neurons), since with a further increase in the number of hidden neurons, the change in the accuracy value is insignificant. Similar studies were carried out on the standard datasets fra-eng (English-French dictionary) and ita-eng (English-Italian dictionary) and similar results were obtained.

Table 1 presents a comparative description of neural networks for translation.

Table 1
Comparative characteristics of neural networks for translation

Network / Criterion	modification Seq2seq	modification of additive attention	modification of multiplicative attention	modification of additive attention with pointing
Accuracy	0.80	0.85	0.90	0.92

Table 1 shows that modified additive attention with pointing has the highest translation accuracy.

5. Conclusions

1. To solve the problem of improving the accuracy of the translation of linguistic structures, the existing methods of neural network translation were investigated. These studies have shown that today the most effective is the use of recurrent neural networks.
2. To improve the quality of the translation of linguistic structures, mathematical models of modified Seq2seq neural networks, additive attention, multiplicative attention, and additive attention with pointing were created. Unlike the corresponding traditional neural networks, in the decoder of the proposed modified neural networks, there are no feedbacks from the output layer to the hidden one, i.e., the decoder's structure coincides with the encoder's structure, which simplifies the practical implementation of the decoder.

3. In the course of a numerical study of neural network translation models, their structure was determined. The experiments performed showed that with 256 hidden neurons (corresponding to the number of neurons for encoding one word), the accuracy value does not change significantly, and the selected network gives translation results with maximum accuracy.
4. The proposed approach can be used in various intelligent systems that use the translation of linguistic structures. For example, in computer systems for supply chain management, where natural language interaction between subjects, which are represented by computer agents, plays an important role.

6. References

[1] R. Aharoni, Y. Goldberg, Towards string-to-tree neural machine translation, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, volume 2: Short Papers, 2017, pp. 132–140. doi: 10.18653/v1/P17-2021.

[2] N. Akoury, K. Krishna, M. Iyyer, Syntactically supervised transformers for faster neural machine translation, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1269–1281. doi: 10.18653/v1/P19-1122.

[3] P. Ghadimi, F. Ghassemi Toosi, C. Heavey, A multi-agent systems approach for sustainable supplier selection and order allocation in a partnership supply chain, European Journal of Operational Research, 269 (2018) 286–301. doi: 10.1016/j.ejor.2017.07.014.

[4] P. Ghadimi, C. Wang, M. Lim, C. Heavey, Intelligent sustainable supplier selection using multi-agent technology: Theory and application for Industry 4.0 supply chains, Computers & Industrial Engineering, 127 (2019) 588–600. doi:10.1016/j.cie.2018.10.050.

[5] A. Swierczek, Decentralization of information and supply chain self-organization: the resulting effect on network performance in the transitive service triads, Supply Chain Management, 28 (2022) 425-449. doi: 10.1108/scm-05-2021-0266.

[6] A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, F. Dalvi, J. Glass, Identifying and controlling important neurons in neural machine translation, in: International Conference on Learning Representations (ICLR), 2019, pp. 1-19. doi: 10.48550/arXiv.1811.01157.

[7] L. H. Baniata, S. Park, S.-B., Park, A multitask-based neural machine translation model with part-of-speech tags integration for Arabic dialects, Applied Sciences 2502 (2018). doi: 10.3390/app8122502.

[8] K.-L. Du, K. M. S. Swamy, Neural Networks and Statistical Learning, Springer-Verlag, London, 2014.

[9] E. Fedorov, V. Lukashenko, V. Patrushev, A. Lukashenko, K. Rudakov, S. Mitsenko, The method of intelligent image processing based on a three-channel purely convolutional neural network, in: CEUR Workshop Proceedings, volume 2255, 2018, pp. 336–351. doi: 10.1109/EWDTS.2013.6673185.

[10] G. G. Shvachych, O. V. Ivaschenko, V. V. Busygin, Ye. Ye. Fedorov, Parallel computational algorithms in thermal processes in metallurgy and mining, Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu, 4 (2018) 129–137. doi: 10.29202/nvngu/2018-4/19.

[11] G. Shlomchak, G. Shvachych, B. Moroz, E. Fedorov, D. Kozenkov, Automated control of temperature regimes of alloyed steel products based on multiprocessors computing systems, Metalurgija, 58 (2019) 299-302.

[12] R. Jozefowicz, W. Zaremba, and I. Sutskever, An Empirical Exploration of Recurrent Network Architectures, in: Proceedings of the 32nd International Conference on MachineLearning, volume 37, 2015, pp. 2342-2350.

[13] A. Wysocki, M. Ławryńczuk, Predictive control of a multivariable neutralisation process using Elman neural networks, in: Advances in Intelligent Systems and Computing, Springer: Heidelberg, 2015, pp. 335-344. doi: 10.1007/978-3-319-15796-234.

[14] M. Berglund, T. Raiko, M. Honkala, L. Kärkkäinen, A. Vetek, J. Karhunen, Bidirectional recurrent neural networks as generative models, in: Proceedings of the 28th International Conference on Neural Information Processing Systems, 2015, pp. 856−864.

[15] M. Sundermeyer, T. Alkhouli, J. Wuebker, H. Ney, Translation modeling with bidirectional recurrent neural networks, in: Proceedings of the Conference on Empirical Methods on Natural Language Processing, 2014, pp. 14-25.
[16] P. Potash, A. Romanov, A. Rumshisky, Ghostwriter: using an LSTM for automatic rap lyric generation, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1919– 1924. doi:10.18653/v1/D15-1221.
[17] B. Cheng, X. Xu, Y. Zeng, J. Ren, S. Jung, Pedestrian trajectory prediction via the Social-Grid LSTM model, in: The 2nd Asian Conference on Artificial Intelligence Technology, volume 2018, no. 16, 2018, pp. 1468–1474. doi: 10.1049/joe.2018.8316.
[18] R. Jin, Z. Chen, K. Wu, M. Wu, X. Li, R. Yan, Bi-LSTM-based two-stream network for machine remaining useful life prediction, IEEE Transactions on Instrumentation and Measurement, 3167778 (2022). doi: 10.1109/TIM.2022.3167778
[19] E. Kiperwasser, Y. Goldberg, Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations, Transactions of the Association for Computational Linguistics 4 (2016) 313–327. doi: 10.1162/tacl_a_00101.
[20] R. Dey, F. M. Salem, Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks, arXiv:1701.05923, 2017. – URL: https://arxiv.org/ftp/arxiv/papers/1701/1701.05923.pdf.
[21] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555, 2014.
[22] S. A. Khan, S. M. D. Khalid, M. A. Shahzad, F. Shafait, Table structure extraction with bi-directional gated recurrent unit networks, in: International Conference on Document Analysis and Recognition (ICDAR), volume 4, No. 2, 2019, pp. 78–88. doi: 10.1109/ICDAR.2019.00220.
[23] M. Artetxe, G. Labaka, E. Agirre, Unsupervised Statistical Machine Translation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3632–3642. doi: 10.18653/v1/D18-1399.
[24] A. H. S. Hamdany, R. R. O. Al-Nima, L. H. Albak, Translating cuneiform symbols using artificial neural network, in: TELKOMNIKA Telecommunication, Computing, Electronics and Control, volume 19, No. 2, 2021, pp. 438-443. doi: 10.12928/telkomnika.v19i2.16134.
[25] M. Artetxe, G. Labaka, E. Agirre, An Effective Approach to Unsupervised Machine Translation, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 194-203. doi: 10.18653/v1/P19-1019.
[26] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1724–1734. doi: 10.3115/v1/D14-1179.
[27] I. Sutskever, O. Vinyals, Q. V. Le. Sutskever I. Sequence to Sequence Learning with Neural Networks, in: Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14), Montreal, Canada, volume 2, 2014, pp. 3104–3112.
[28] D. Bahdanau, K. Cho, Yo. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, in: International Conference on Learning Representations, 2015, pp.1-15. doi: 10.48550/arXiv.1409.0473
[29] M.-Th. Luong, H. Pham, Ch. D. Manning, Effective Approaches to Attention-based Neural Machine Translation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1412–1421.
[30] A. See, P. J. Liu, , C. D. Manning, Get to the point: Summarization with pointer-generator networks. arXiv:1704.04368v2, 2017. doi:10.48550/arXiv.1704.04368.