УДК 621.395.721.5
DOI: https://doi.org/10.17721/AIT.2023.1.06

Tania STAROVOYT, Student
ORCID ID: 0009-0008-6335-7679
e-mail: starovoyt.tania@lll.kpi.ua
The National University of Water and Environmental Engineering, Rivne, Ukraine

Yuriy ZAYCHENKO, DSc (Engin.), Prof.
ORCID ID: 0000-0001-9662-3269
e-mail: zaych@i.com.ua
National Technical University of Ukraine Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine

A HYBRID QUANTUM-PERFECTED MODEL OF ARTIFICIAL INTELLIGENCE IN THE PROBLEM OF AUTOMATIC RECOGNITION AND FAST CONVERSION OF UNSTRUCTURED TEXT INFORMATION INTO SPATIAL

B a c k g r o u n d . ***Efficiently converting large amounts of unstructured text data into spatial information is crucial for managing water distribution systems. This allows for the conversion of extensive sets of text information, such as reports, orders, letters, and other documents, into point classes of spatial objects in geographic information systems. To tackle this challenge, a promising new approach involves combining hybrid quantum-classical neural networks with geo-information technologies.***

M e t h o d s . ***The study utilized quantum-enhanced hybrid neural networks in combination with GIS methods to identify named entities such as personal accounts and balance sheet objects of Kyivvodokanal by their addresses and geocoding. This information was then published on a geoportal using the ArcGIS Enterprise platform in real-time, which holds great promise for effective water management. The performance of the developed model was evaluated by accuracy indicators, recall parameters, and weighted harmonic average of accuracy and recall.***

R e s u l t s . ***The obtained results indicate that the developed hybrid quantum-classical model of artificial intelligence can be successfully applied to transform large volumes of unstructured textual information into spatial information. The model was integrated into GIS using ArcGIS Enterprise. By combining the obtained point classes of spatial objects with already existing data, methods of spatial connections, an interactive map with an update interval of every five minutes was developed.***

C o n c l u s i o n s . ***Taking advantage of quantum computing and combining it with classical hardware and classical AI models, it became possible to achieve similar and even better performance in various tasks compared to state-of-the-art methods. Quantum natural language processing is a promising new field that has the potential to revolutionize the way one analyzes and understands human language.***

K e y w o r d s : ***QNLP, hybrid quantum-classical neural networks, geocoding, geoparsing, geographic information systems, ArcGIS, entity recognition, unstructured textual information, variational quantum igniter, QLSTM, spatial objects.***

Background

Efficiently converting large amounts of unstructured text data into spatial information is crucial for managing water distribution systems. This allows for the conversion of extensive sets of text information, such as reports, orders, letters, and other documents, into point classes of spatial objects in geographic information systems. To tackle this challenge, a promising new approach involves combining hybrid quantum-classical neural networks with geo-information technologies. Quantum-classical artificial neural networks offer several advantages over classical computing (Rieffel, & Polak, 2011), including computing based on the superposition of quantum states and entanglement for the correlation of quantum states. Most of the unstructured data contains location-based entities, making it spatial. Additionally, text information related to water supply companies, such as reports, orders, personal accounts data, and balance sheet reports, is also spatial. This is because it refers to objects situated underground or on the surface of a particular territory, such as pipelines, pumping stations, shut-off valves, etc. These objects have a coordinate reference, and the data may contain a specific address or object name, such as a shopping center or museum. Geoparsing methods can be used to extract the coordinates of an address, which may include famous objects at the end of a street.

Natural language processing (NLP) is a subfield of artificial intelligence that focuses on the study of human language by studying the meanings of words and sentences (DisCoCat lambeq). Methods of combining GIS and NLP were studied in works (Lawley et al., 2023; Ma et al., 2021; Enkhsaikhan et al., 2021a; Enkhsaikhan et al., 2021b). Language models contain mathematical rules for representing words, parts of words, or sentences as numerical vectors that are used to solve machine learning problems (Enkhsaikhan et al., 2021b).

The most effective NLP models are based on transformers, and deep learning models that can interpret the meaning of words based on context (Floridi, & Chiriatti, 2020). Transformers and self-attention are used to capture long-term and bidirectional dependencies, which are important for representations of words that have multiple meanings.

There are two main categories of research methods for recognizing named entities: those based on spatial statistics and machine learning, and those that use deep learning techniques. The former involves analyzing patterns of spatial distribution of toponyms in the text. De Bruijn et al. have suggested a method for extracting toponyms by comparing databases of toponyms and OpenStreetMap (Airola et al., 2019). Current methods that rely on specific geographic directories are unable to identify unofficial place names mentioned in unstructured text. In an effort to address this issue, McKenzie et al utilized numerical spatial statistical metrics and random forest learning techniques to extract neighborhood names from rental property listing data (De Bruijn et al., 2019). Another study conducted by Lai et al utilized a method based on spatial point pattern analysis to extract toponyms from geotagged Twitter texts (McKenzie et al., 2018). However, these methods are limited to specific research regions and may encounter issues such as high dimensionality, computational complexity, and the need for frequent retraining.

To reduce computational complexity, one way of extracting toponyms is by using named entity recognition (NER) tools that already exist. Hu et al. utilized NER tools like spaCy NER and Stanford NER to extract toponyms from real estate ads, which were then filtered using spatial clustering (Lai et al., 2020). However, this method has limited generalizability. Alternatively, artificial neural networks are great for addressing issues of generality and scalability. However, basic neural network structures may struggle with adapting to large-scale corpora.

Quantum Natural Language Processing (QNLP) is a recently developed area that combines the principles of quantum mechanics with natural language processing techniques. This combination allows for more efficient and accurate analysis of human speech compared to traditional NLP methods by utilizing unique quantum mechanics capabilities such as superposition and entanglement (Bhagvan, 2020; Guarasci, De Pietro, & Esposito, 2022; Hoffmann, 2021). Quantum algorithms further enhance the process by performing multiple calculations simultaneously. Overall, Quantum Natural Language Processing shows great potential in revolutionizing the way we analyze and comprehend human language.

The current solutions for named entity recognition and GIS integration have some shortcomings, such as being slow and not always accurate. As a result, a new hybrid model of artificial intelligence was developed to improve the process and results. Our study proposes a to recognize named entities and convert the information into spatial data. The method of integrating quantum processing of natural language to geo-information systems is carried out for the first time with sets of text information of Kyivvodokanal and for the first time uses quantum computing to improve the classical long-short-term memory (LSTM) artificial neural network model.

Methods

This paper uses a combination of a number of methods. These methods include method of integrating quantum processing of natural language to geo-information systems, method of geoparsing and geocoding and method of quantum state coding. Let us consider them in their turn.

Development methodology and model training workflows. In this study, we utilized text file sets containing personal accounts from Kyivvodokanal subscribers, as well as unstructured text files containing information on the ownership of objects in the balance sheet. Orders for water supply networks and structures transfer to the possession and use of PJSC "Kyivvodokanal" provided crucial information for the GIS water supply system, including the date, transfer type, number, name of structural unit, network type, network length, pipe material, year of commissioning, and diameter. Personal accounts contain information: account number, type of input, address, place of installation of input, source of water use, volume of consumption, etc. The data were received in text formats txt, CSV.

To identify addresses and other objects in text files, we used an open-source tool called Doccano (Qiskit, 2023) for text labeling in natural language processing (NLP) tasks. We categorized all relevant information into pre-defined categories, such as names of people, organizations, and locations. Each entity tag was encoded using the VIO annotation scheme, where the letter "B" represented the beginning of the entity tag, and "I" represented the end. Any words that didn't refer to the desired objects were marked with the "O" tag.

Named Entity Recognition (NER). Named Entity Recognition (NER) is the process of extracting or identifying key information (entities) in text. An entity can be a keyword or a series of keywords. NER is a form of natural language processing (NLP), a subfield of artificial intelligence. Entities in NER are divided into the following categories: person, organization, time, location, and others. This model is used when there is a need to group texts based on their correspondence or similarity (Peixeiro, 2022). The entity recognition process is divided into entity recognition model training and object recognition using deep learning.

1. The process of training an entity recognition model involves using a machine learning approach to provide the model with training samples. These samples consist of pairs of input text and labeled entities in the text. The training process is iterative, with the training data passing through the neural network multiple times. Each pass is called an epoch (Peixeiro, 2022).

2. Object recognition using deep learning. The trained model recognizes named entities and locations in text files and records them in a table by them with the specified locator. As a result, we get a point class of spatial objects.

To run a deep learning model, it's important to indicate whether the base layers in the pre-trained model will be fixed to keep their original weights and biases. By default, the base model layers are not frozen, allowing the weights and offsets of the reference model to adjust to the training samples. This may take longer, but usually yields better results.

Geoparsing and geocoding. Geoparsing is a process of toponym recognition and geolocation that converts arbitrary text descriptions of places into spatial identifiers (coordinates) from structured or unstructured text, such as social media posts, news, scientific articles, web pages, and historical archives. Geoparsing is used for the following typical areas: searching for geographic information, combating natural disasters, traffic management, fighting crime, and monitoring diseases.

The process of geoparsing involves several main procedures (Wang et al., 2022).

1) General procedure, which includes recognizing place names and determining their locations. Various tools like StanfordNER (Doccano), SpaCyNER (The Stanford Natural Language Processing Group), and Baidu geocoding tool (spaCy) are often used in this stage. The goal is to use modern text-based geoanalysis methods to obtain the most accurate geographic analysis locations.

2) The procedure for associating attributes. To associate attributes with toponyms, the procedure involves extracting attributes from the sentence and pairing them with the corresponding toponyms. The aim of this process is to determine the primary connections between the toponyms and their attributes.

3) RS function association procedure. The RS function association procedure involves linking attributes with RS functions. These functions can offer extra spatial information to accurately analyze geographical locations.

4) Binding procedure of the RS region. Acquisition of target spatial information based on RS characteristics is performed.

5) Location correction procedure. The resulting locations and references are synthesized, including “places” from the general geoanalysis procedure, “attribute toponym” from the attribute association procedure, “attribute-RS function” from the feature association procedure RS, and “target-area function-RS. On the example of the study of the ecological model of the forest (Wang et al., 2022), a set of toponyms with geoparsing $t_i \in T$ indicates a toponym-attribute correspondence list. $List_{t\text{-}p}$ indicates the matching list of the RS attribute function. $List_{p\text{-}ft}$ set of zone RS means $fa_j \in A$. Additional parameters are also needed to improve correction accuracy, including a list of the city center $List_{cc}$. The cluster threshold divides the target areas into clusters. To determine the correction of toponyms, it is necessary to calculate the value of AI, which indicates the ability to attract the surrounding territories. The value of AI is calculated according to formula (1) (Wang et al., 2022).

$$AIk = AREAck / DISTANCEck. \quad (1)$$

Geocoding is the process of converting addresses into geographic point objects with coordinates. Reverse geocoding converts coordinates to addresses. Unlike geoparsing, geocoding parses unambiguous structured references to locations, such as postal addresses and well-formatted numeric coordinates. For our research, we used the Google Geocoding API, as it is free for processing a large number of objects and works well for non-English addresses.

Quantum state coding method. To encode data in a quantum state, a basic language set of tokens is defined for each representative space of token values (a noun is a subject, verbs and nouns are objects). The generalized representation of each lexeme t_i in the corresponding value space will be defined as m (O'Riordan et al., 2020).

$$t_i = \sum_j^n d_{i,j} m_j \ , \quad (2)$$

where $d_{i,j}$ means the distance between the base marker m_j and the non-base marker t_i. As a result, we get a linear combination of base tokens with representative weights to describe the matched tokens. For effective data preprocessing, the following steps must be performed (O'Riordan et al., 2020):

1) Tokenize the corpus and record the places of occurrence in the text.
2) Mark lexemes with the appropriate type of value space.
3) Divide lexemes into sets of nouns and verbs.
4) Identify the main markers in each set as the most frequently occurring lexemes.
5) Map base markers in each corresponding space on a fully connected graph with edge weights determined by the minimum distance between base markers.
6) Calculate the shortest Hamilton cycle. The order of tokens in a cycle reflects the division of tokens in the text and the degree of their similarity.
7) Map main markers to binary strings.
8) Project the composite tokens onto the base set of tokens using the limit distances of the similarity representation.
9) Create a sentence by matching compound lexemes noun-verb-noun. Once the pre-processing stages are complete, the corpus is converted into binary strings and then encoded into the quantum register (O'Riordan et al., 2020). To make the coding process more straightforward, we use a binary representation of the distance for formula (3), where markers within the same cutoff have identical weight. This enables states to be encoded as an evenly weighted superposition, which can be easily implemented as a quantum circuit (O'Riordan et al., 2020).

For the coding algorithm of a set of bit strings, we define: basic states of calculations and, Pauli-X gate (σ_x), controlled X and n-controlled NOT operations (nCX), respectively (O'Riordan et al., 2020):

$$\mathrm{X}a : |a\rangle \rightarrow |\neg a\rangle \ , \quad (3)$$

$$\mathrm{CX}a,b : |a\rangle|b\rangle \rightarrow |a\rangle\, |a \oplus b\rangle, \quad (4)$$

$$\mathrm{nCX}a_1, b \ldots a_n, b : |a_1\rangle \ldots |a_n\rangle\, |b\rangle \rightarrow |a_1\rangle \ldots\ |a_n\rangle\ \ |b\ \oplus (a_1 \wedge \ldots \wedge a_n)\rangle\ . \quad (5)$$

For a set of N unique binary patterns, each of length n, three qubit registers are required, a memory register of length n, an auxiliary register of length n, and a control register of length 2, the complete quantum register is determined by the formula (O'Riordan et al., 2020):

$$p^{(i)} = \left\{p_1^{(i)}, \ldots, p_n^{(i)}\right\} i = 1, \ldots, N|m|a\rangle|u\rangle|01\rangle|m\rangle|a\rangle, \quad (6)$$

$$|m\rangle = \ |a\rangle = \ |0\rangle^{\oplus n}, \quad (7)$$

$$|\psi_0\rangle = \ |a\rangle|u\rangle|m\rangle = |0_1 \ldots 0_n\rangle|01\rangle|0_1 \ldots 0_n\rangle\ . \quad (8)$$

Each of the binary vectors is encoded sequentially. In each iteration of the coding algorithm, a new state is generated in the superposition. The new state is called the active state of the next iteration. The remaining states are inactive. In each iteration of the algorithm, the active state is selected using $|u\rangle = |01\rangle$ (O'Riordan et al., 2020).

During the execution of one iteration, the binary vector is stored in an integer format, which is sequentially encoded bit by bit using Pauli-X elements into an auxiliary register that causes the state: $|a\rangle|\psi\_1\rangle$ (Andrade et al., 2020).

$$|\psi_1\rangle = |a_1^{(1)} \ldots\ a_n^{(1)}\rangle|01\rangle|0_1 \ldots 0_n\rangle. \quad (9)$$

In the next step, this binary representation is copied into the active state memory register by using the $2CX$:$|m\rangle|1\rangle$ gate (O'Riordan et al., 2020).

$$|\psi_2\rangle = \ \prod_{j=1}^n 2CX_{aju2m_j}|\psi_1\rangle. \quad (10)$$

The next step is to apply CX and then X-gate to all qubits, using qubits as control elements

$$|\psi_3\rangle = \ \prod_{j=1}^n X_{mj} CX_{a_j m_j}|\psi_2\rangle. \quad (11)$$

This step sets the qubits to 1, provided that the corresponding qubit index in both matches, otherwise 0. Therefore, the state in which the register matches the stored pattern will be set to 1, while the other states will have at least one 0. After the state is selected, the nCX operation is applied to the first qubit in the auxiliary register, using the qubits as control: $|m\rangle$ (O'Riordan et al., 2020).

$$|\psi_4\rangle = \ nCX_{m_1 \ldots m_n u_1}|\psi_3\rangle. \quad (12)$$

To fill a new state in the superposition, it is necessary to remove the amplitude from the existing states so that the new state has a non-zero coefficient. For this, it is necessary to use the controlled unitary matrix $CS^{(N+1-i)}$ to the second auxiliary qubit u_2, where the first auxiliary qubit u_1 is the control (O'Riordan et al., 2020):

$$|\psi_5\rangle = \ CS_{u_1 u_2}^{(N+1-i)}|\psi_4\rangle\ . \quad (12)$$

ISSN 2788-6603

All other states are selected by the formula:

$$k \in \{1, \dots, N\} \subset Z^{+} R_y(\Theta) = \exp\,(-i\Theta\sigma_y/2),$$

$$\gamma(k) = -\arccos \arccos\,((k-2)/k\;|u\rangle = |11\rangle|u\rangle = |10\rangle|u\rangle = |00\rangle. \tag{13}$$

For the next iteration of the algorithm, we calculate the steps from formulas (10) (11) (12) as (O'Riordan et al., 2020):

$$|\psi_6\rangle = nCX_{m_1 \dots m_n u_1}|\psi_5\rangle, \tag{14}$$

$$|\psi_7\rangle = \prod_{j=n}^{1} CX_{a_j m_j} X_{m_j}|\psi_6\rangle, \tag{15}$$

$$|\psi_8\rangle = \prod_{j=n}^{1} 2CX_{a_j u_2 m_j}|\psi_7\rangle. \tag{16}$$

This result leads to the selection of the previous active state $|u\rangle = |00\rangle$, while the new state will be $|u\rangle = |01\rangle$. The memory register of the previous active state will contain the pattern $\{a^{(i)}, \dots, an^{(i)}\}$, while the memory register of the new active state is set to all zeros (O'Riordan et al., 2020).

The next step is to code the quantum register for another pattern, according to formula (17) (O'Riordan et al., 2020):

$$|\psi\rangle = |a\rangle|u\rangle|m\rangle = |0_1 \dots 0_n\rangle|00\rangle\left(\frac{1}{\sqrt{N}}\sum_{i=1}^{N}|p_1^{(i)}, \dots, p_n^{(i)}\rangle\right). \tag{17}$$

This algorithm assumes that the number of patterns to be encoded is known in advance. These templates are necessary for generating $S^{(k)}$ matrices and applying them in the correct order. The total number of qubits for this algorithm is $2n + 2$, of which $n + 2$ can be reused (O'Riordan et al., 2020).

To illustrate how one can apply above methods, let us apply them to a number of neural networks.

First, we will consider hybrid quantum-enhanced neural network (QNN). A quantum neural network takes classical input data and maps it into a quantum state using a feature map. The resulting state passes through the variational form. The output data of a quantum neural network is the result of the final state measurement operation (Gonzlez-Castillo, & Combarro, 2023). Fig. 1 schematically shows how a quantum neural network works.

QNNs, like classical artificial neural networks, can be trained using backpropagation of error with gradient descent (differentiated learning) or evolutionary search heuristics (undifferentiated) (Jacquier et al., 2022). Quantum neural networks have an edge over classical ones when it comes to quantum data. However, classical data can also be encoded as quantum states and processed by QNNs with more efficiency and better ability to handle overtraining due to their speedup (Jacquier et al., 2022).

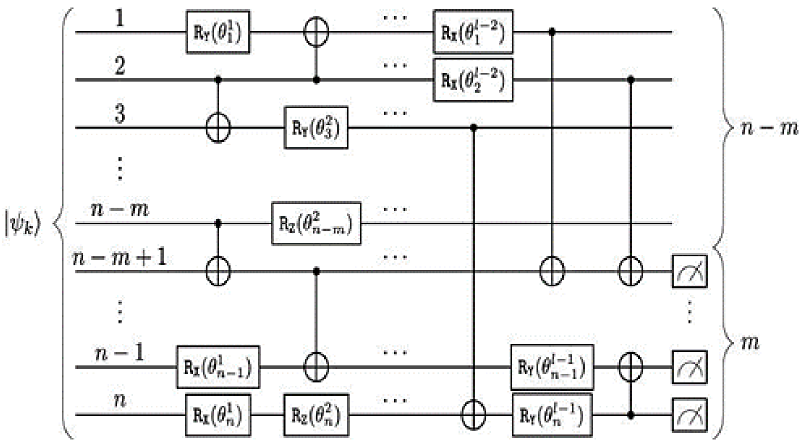

Fig. 1. Schematic representation of a quantum neural network (Jacquier et al., 2022)

When quantum neural networks work on simulators, the gradients can be calculated using automatic differentiation methods. When running either on real quantum equipment or on simulators, the gradients can also be calculated using the parameter shift rule (Gonzlez-Castillo, & Combarro, 2023). These methods are fully implemented in the PennyLane quantum programming environment (Baidu).

There are optimization methods called "optimizers without gradients" that eliminate the need for calculating gradients during the backpropagation step (Zhu et al., 2019). This approach reduces the complexity of differentiating quantum circuits. It was frequently used in the creation of the initial QNNs.

Quantum Natural Language Processing (QNLP). Quantum Natural Language Processing (QNLP) is a new field of quantum artificial intelligence where natural language is combined with the capabilities of quantum computing. QNLP is a new line of research aimed at developing and improving NLP models using quantum phenomena: superposition, quantum entanglement, and interference in tasks related to natural language processing, encoding information into a quantum state, or using quantum equipment (Guarasci, De Pietro, & Esposito, 2022).

Quantum gates are used to manipulate quantum states, which are similar to logic gates in classical computing. QNLP algorithms utilize these gates to perform NLP tasks like classification, clustering, and generation by applying them to the quantum states that represent words and sentences (Guarasci, De Pietro, & Esposito, 2022).

Long short-term memory (LSTM) networks. Long Short-Term Memory (LSTM) is a deep learning architecture that is a subtype of RNN. LSTM solves the short-term memory problem by adding cell state. This allows past information to pass through the network over a longer period, meaning the network is still carrying information from early values in the sequence. Long-short-term memory (LSTM) networks have been a breakthrough in natural language processing when learning sequential data and dependencies, such as textual data. Transformer architectures have outperformed LSTMs, but long-short-term memory networks can be greatly improved by applying quantum computing, which in turn outperforms Transformer models. Fig. 2 shows the LSTM architecture.

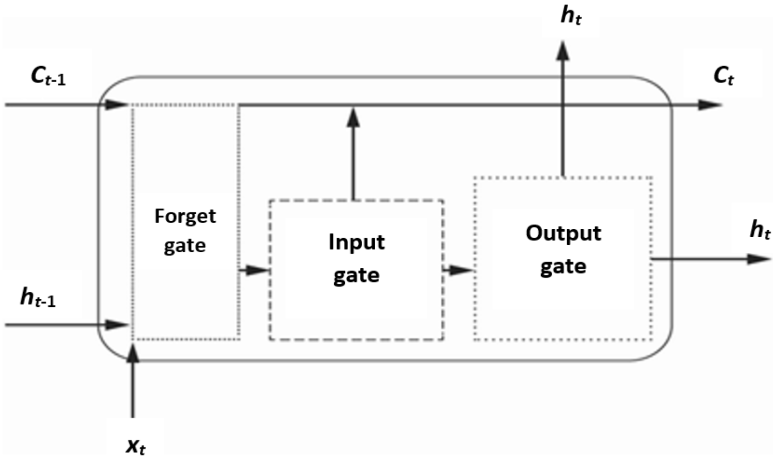

Fig. 2. LSTM Architecture

Adding a cell state labeled *C*. This cell state allows the network to store past information in the network for a longer time, solving the problem of gradient fading. The element of the sequence is indicated by x_t, the hidden state is h_t. The state of the cell and the hidden state are passed to the next element of the sequence, using the previous information as input for the next element (Chen, 2022).

Quantum-Enhanced Network of Long Short-Term Memory (QLSTM). The QLSTM quantum long-term memory network is an advanced version of the classical LSTM that operates in the quantum realm. It replaces LSTM cells with variational quantum circuits (VQC), which have three components: data encoding, variational layer, and quantum measurements (Wu, & Wang, 2019). This circuit converts a classical input vector into a quantum state, and the process is outlined in section "Quantum state coding method". The mathematical model for QLSTM is presented in formulas (18–24) and Fig. 3.

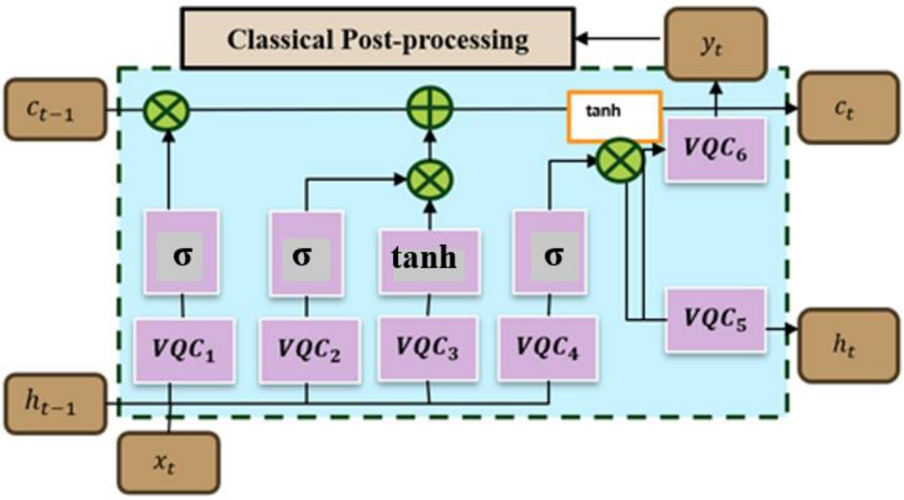

Fig. 3. QLSTM Architecture scheme (Wu, & Wang, 2019)

$$f_t = \sigma(VQC_1\,(v_t)), \tag{18}$$
$$i_t = \sigma(VQC_2\,(v_t)), \tag{19}$$
$$\widetilde{C}_t = \tanh(VQC_3\,(\,v_t)), \tag{20}$$
$$c_t = f_t * c_{t-1} + i_t * \widetilde{C}_t, \tag{21}$$
$$o_t = \sigma(VQC_4\,(v_t)), \tag{22}$$
$$h_t = VQC_5(o_t * \tanh\,(c_t)), \tag{23}$$
$$y_t = VQC_6(o_t * \tanh\,(c_t)). \tag{24}$$

ISSN 2788-6603

The blocks σ and tanh are the activation functions of the sigmoid and the hyperbolic tangent. Variable x_t is the input signal at time t, h_t is for the hidden state, c_t is for the cell state, y_t is the output. ⊗ and ⊕ are element-wise multiplication and addition, respectively (Wu, & Wang, 2019).

Metrics for evaluating the accuracy of models. To evaluate the accuracy of the models, the accuracy_score formula (25) was used. If all the predicted labels for a sample match the true set of labels, the subset accuracy is 1.0. Otherwise, it is 0.0. The proportion of correct predictions over the n_{samples} is defined as (PennyLane, 2023), where y_i is the predicted value and the corresponding true value is y_i.

$$\text{Accuracy}\ (y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i), \quad (25)$$

where 1(*x*) is the identity function.

The F-measure is a weighted harmonic mean of precision and recall scores. It is defined as (PennyLane, 2023):

$$\text{AP} = \sum_n (R_n - R_{n-1}) P_n, \quad (26)$$

where P_n and R_n are precision and recall at the nth threshold. With random predictions, AP is the proportion of positive samples (PennyLane, 2023).

The Matthews correlation coefficient takes into account false positives and negatives, and is considered a balanced measure that can be used even if the classes are of different sizes (PennyLane, 2023). For the research task, that is, for multi-class classification, the Matthews coefficient is determined by the formula:
$t_k = \sum_i^K C_{ik}$ – number of classes, $p_k = \sum_i^K C_{ki}$ – number of predicted classes, $c = \sum_k^K C_{kk}$ – total number of correctly predicted samples, $s_k = \sum_i^K \sum_j^K C_{ij}$ – total number of samples. Then we define a multi-class MCC as (PennyLane, 2023):

$$\text{MCC} = \frac{c \times s - \sum_k^K p_k \times t_k}{\sqrt{(s^2 - \sum_k^K p_k^2) \times (s^2 - \sum_k^K t_k^2)}}. \quad (27)$$

If there are more than two label values, the MCC value will no longer be in the range of –1 to +1. Instead, the minimum value will be somewhere between –1 and 0, depending on the number and distribution of ground truth labels. The maximum value will always be +1 (PennyLane, 2023).

The *ROC-AUC* or *AUROC* function, known as roc_auc_score, calculates the area under the ROC curve and reduces all the curve information to a single number. This function is also applicable to multi-class classification. It supports two averaging strategies: the one-versus-one algorithm, which averages pairwise *ROC AUCs*, and the one-versus-rest algorithm, which averages ROC AUCs for each class against all other classes. In both cases, the predicted labels are provided in an array with values ranging from 0 to *n*_classes, while the scores reflect the estimated probability that the sample belongs to a particular class (PennyLane, 2023).

The one-versus-one algorithm calculates the average *AUC* of all possible pairwise combinations of classes. The evenly weighted multi-class AUC metric is determined by the formula (PennyLane, 2023):

$$\frac{1}{c(c-1)} \sum_{j=1}^{c} \sum_{k>j}^{c} (AUC(k) + AUC(k|j)), \quad (28)$$

where *c* is the number of classes and $AUC(k)$ is positive class j and k is negative.
In the general multiclass case $AUC(k) \neq AUC(j)$ (PennyLane, 2023).

Results

LSTM model implementation. To implement the classic LSTM, we used the Google Colab and PyTorch development environment. The first step was to configure the appropriate packages and load a set of labeled data, with sentences padded with zeros to the same length. Next, we defined two functions: train_model – a function for training the model; val_model - to check the model. The classic LSTM model consists of four layers: Forget layer, Input layer, Update layer, Output layer. A learning rate of 0,05 was determined after several experiments with different amounts of data. The results of model training are shown in Fig.4 and Tab.1.

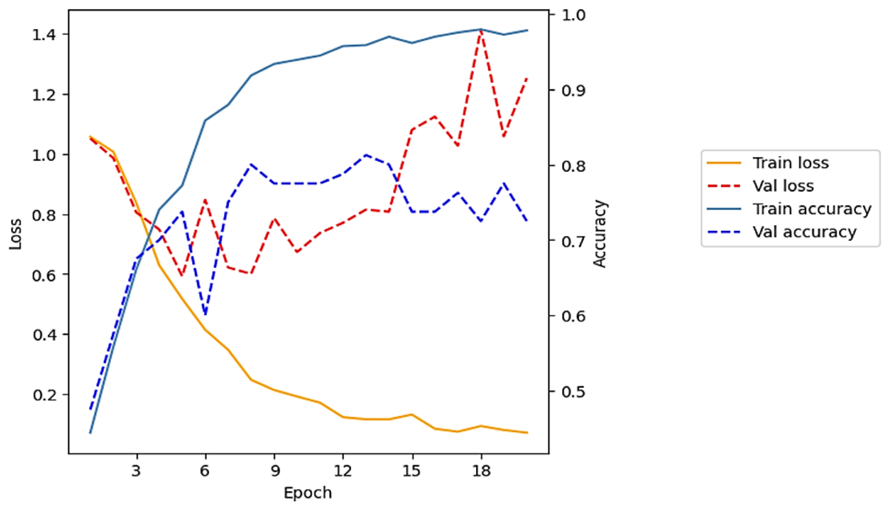

Fig. 4. The result of learning the classic LSTM model

Table 1

Results of classical LSTM network training

Epoch	LSTM			
	Train data		Valid data	
	loss	acc	loss	acc
1	1,056	0,44	1,051	0,47
2	1,006	0,56	0,986	0,57
3	0,836	0,66	0,805	0,68
4	0,629	0,74	0,748	0,70
5	0,517	0,77	0,592	0,74
6	0,414	0,86	0,846	0,60
7	0,348	0,88	0,621	0,75
8	0,248	0,92	0,600	0,80
9	0,214	0,93	0,786	0,77
10	0,192	0,94	0,673	0,77
11	0,172	0,94	0,736	0,77
12	0,124	0,96	0,770	0,79
13	0,117	0,96	0,814	0,81
14	0,116	0,97	0,807	0,80
15	0,132	0,96	1,079	0,74
16	0,085	0,97	1,123	0,74
17	0,075	0,98	1,026	0,76
18	0,094	0,98	1,413	0,73
19	0,081	0,97	1,058	0,77
20	0,072	0,98	1,251	0,73

LSTM compilation was performed using sparse categorical cross-entropy loss and accuracy metrics. From the obtained results, it can be seen that the train loss is constantly decreasing, which indicates that the network is learning well. Accuracy of the model: 0.76, average harmonic means of accuracy and recall – 0.771, AUC – 0.823, MCC – 0.616.

Implementation of the QLSTM model. To implement QLSTM, we replaced the key levels of the LSTM neural network with variational quantum layers based on variational quantum schemes (Fig. 5). Thus, the parameters to be learned became the rotation parameters in the chains, forming a hybrid quantum-classical level of the neural network (as a classical optimizer).

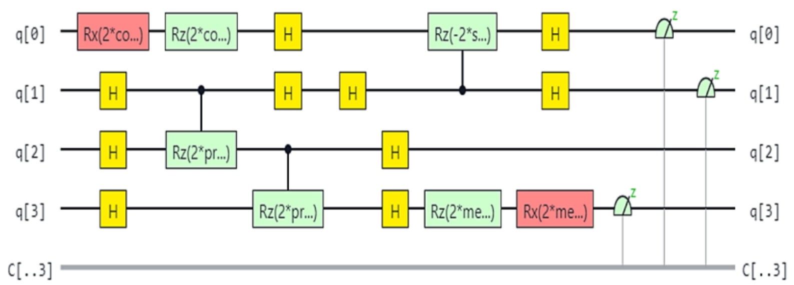

Fig. 5. Variational quantum scheme

The built-in Pennylane quantum simulator [Baidu] was used to run variational quantum circuits. The number of qubits we used is 4, the number of variational layers is 1, and the learning rate is 0.05. These parameters were determined by experimenting on a small number of epochs to determine the optimal number that would give the best results. The results of model training are shown in Fig. 6 and Tab. 2.

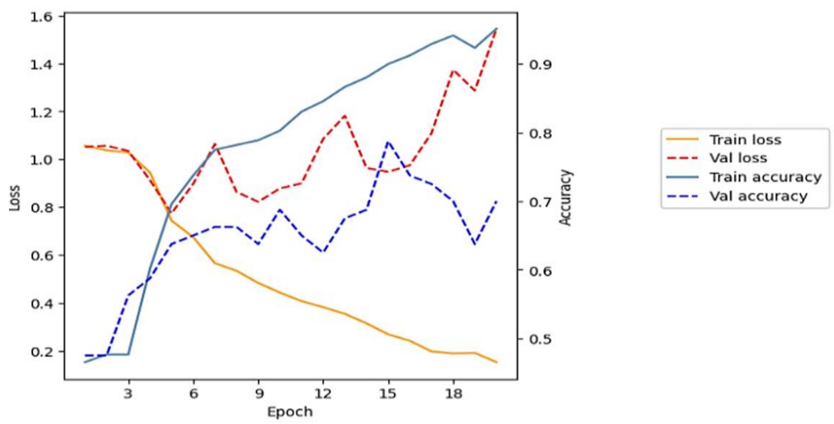

Fig. 6. Results of QLSTM network training

ISSN 2788-6603

Table 2

Results of QLSTM model training

Epoch	QLSTM			
	Train data		Valid data	
	loss	acc	loss	acc
1	1,056	0,47	1,051	0,47
2	1,037	0,48	1,056	0,47
3	1,028	0,48	1,035	0,56
4	0,945	0,60	0,912	0,59
5	0,743	0,70	0,774	0,64
6	0,674	0,74	0,898	0,65
7	0,565	0,77	1,063	0,66
8	0,534	0,78	0,863	0,66
9	0,483	0,79	0,822	0,64
10	0,443	0,80	0,878	0,69
11	0,407	0,83	0,899	0,65
12	0,382	0,85	1,085	0,62
13	0,354	0,87	1,182	0,68
14	0,314	0,88	0,963	0,69
15	0,268	0,90	0,947	0,79
16	0,241	0,91	0,975	0,74
17	0,197	0,93	1,108	0,73
18	0,189	0,94	1,374	0,70
19	0,190	0,92	1,287	0,64
20	0,152	0,95	1,545	0,70

The obtained results showed that hybridization of classical LSTM did not bring significant results, so we additionally performed hybridization of LSTM using categorical compositional distribution model (DisCoCat) (Hu, Mao, & McKenzie, 2019; Scikit-learn), which combines vector space models and categorical compositional models (grammars). DisCoCat is a distributed correspondence of categorical quantum mechanics that provides a way to represent linguistic meanings as quantum processes. In this framework, words and sentences are represented as quantum states, and connections between words are modeled as quantum processes.

The above approach is suitable for small amounts of data because it lacks memory and other advanced features. Therefore, we used DisCoCat charts for LSTM hybridization. This method is well suited for scaling and processing large sequences, as it can transfer previous memory to subsequent processing steps.

The learning results of the hybrid quantum- enhanced neural network are shown in Fig. 7 and Tab.3.

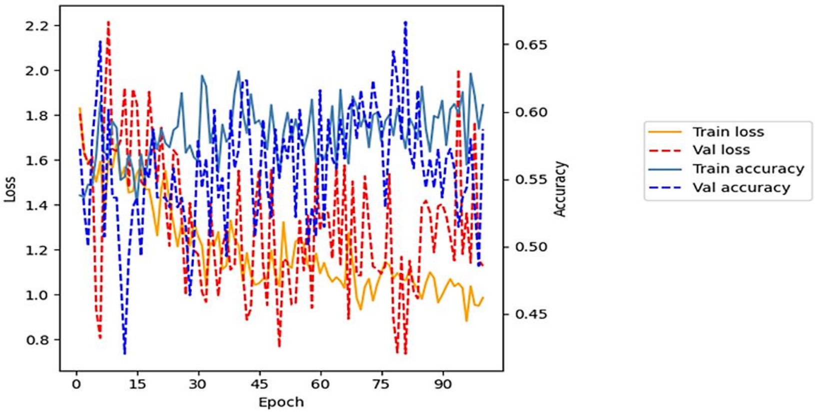

Fig. 7. Results of QLSTM model training

Table 3

DisCoCat model training speed and accuracy results

Epoch	DisCoCat			
	Train data		Valid data	
	loss	acc	loss	acc
1	0,7780	0,5429	1,8223	0,3000
2	0,4895	0,5000	0,6300	0,6667
3	1,0138	0,7143	0,7838	0,6667
4	0,4800	0,7143	0,8864	0,5667
5	0,5569	0,6857	0,8657	0,4667
6	0,9278	0,6429	1,0460	0,5667
7	0,8512	0,5286	0,8624	0,5333
8	0,8151	0,5429	0,8426	0,5000

Epoch	DisCoCat			
	Train data		Valid data	
	loss	acc	loss	acc
9	0,8157	0,4857	1,0738	0,6333
10	1,1278	0,4714	0,8331	0,6000
11	0,8942	0,5000	0,6632	0,6333
12	0,4706	0,5571	1,0278	0,4333
13	0,8645	0,5286	0,8895	0,6333
14	0,9549	0,5286	1,1223	0,5333

As can be seen from the obtained results, the hybridization of the LSTM network with DisCoCat diagrams did not outperform the classical model either.

Accuracy assessment and comparison of LSTM and QLSTM-DisCoCat models. The accuracy of the model was evaluated by the accuracy score (accuracy_score) and the loss function on the training and test data.

From the obtained results, we can see that the results of the QLSTM and DisCoCat models are not superior to the classical model, and the quantum neural network required more training time. With the increase in the number of epochs, the time for training the models increased by a factor of three. The result of changing the number of training epochs is the conclusion that this parameter is variable and depends on the data set used.

It should be noted that when the number of quantum qubits increases, the time for training models increases depending on the size of the data set. The number of qubits in a neural network is a parameter that determines the number of connections when connecting classical layers, a pre-trained model and a quantum layer. The result of changing the number of quantum qubits is the conclusion that this parameter is important for improving the results of the neural network and does not depend on the data set used.

As a next step, we significantly increased the amount of input data and verified the obtained results. The results of model training are shown in Tab. 4.

From the obtained model training results (Tab. 4.), we can see that the hybrid quantum-classical neural network learns much more information in the first few epochs compared to LSTM, but both models eventually degrade to a low value.

Table 4

Results of model training

Epoch	LSTM		QLSTM	
	Train loss	Valid acc	Train loss	Valid acc
1	0,265	2,634	0,007	1,317
2	0,034	2,006	0,006	1,251
3	0,018	1,789	0,005	1,194
4	0,012	1,614	0,004	1,141
5	0,009	1,514	0,004	1,078
6	0,008	1,481	0,004	1,042
7	0,008	1,418	0,004	1,022
8	0,007	1,410	0,004	0,995
9	0,006	1,378	0,003	0,987
10	0,006	1,322	0,003	0,968
11	0,006	1,304	0,003	0,951
12	0,005	1,284	0,003	0,928
13	0,005	1,315	0,003	0,939
14	0,005	1,334	0,003	0,929
15	0,005	1,264	0,003	0,937
16	0,005	1,287	0,003	0,928
17	0,004	1,289	0,003	0,910
18	0,004	1,224	0,003	0,905
19	0,004	1,293	0,003	0,910
20	0,004	1,269	–	–

This indicates that QLSTM has better learning ability. It should also be noted that QLSTM uses fewer parameters than classical LSTM, which also indicates an advantage in the learning ability of QLSTM over LSTM.

Integration of a learned quantum natural language processing model into GIS. To implement the trained model of artificial intelligence in practice, we used ArcGIS Pro, ArcGIS API for Python, in which it is possible to extract objects and record them in a spatial Data Frame or a class of spatial objects, with their simultaneous visualization on the map. Spatial Data Frame (SEDF) creates a simple object that can easily manipulate geometric and attribute data (DisCoCat lambeq).

SEDF is based on data structures that are inherently suitable for data analysis with operations to filter and validate subsets of values that are fundamental to statistical and geographic manipulation (DisCoCat lambeq). We integrated the code of the QLSTM model into the ArcGIS Pro environment to use the already trained entity recognizer model.

To transform the received text information into spatial information, a locator was used, which uses information about the location zone found in text documents to identify the research region in which the addresses are located. As a result, we received a point class of spatial objects with recognized information and published them on the geoportal for further geoprocessing (Fig. 8).

Fig. 8 shows the result of converting text into spatial information (point classes of spatial objects) and publishing the resulting data on a corporate geoportal based on ArcGIS Enterprise.

ISSN 2788-6603

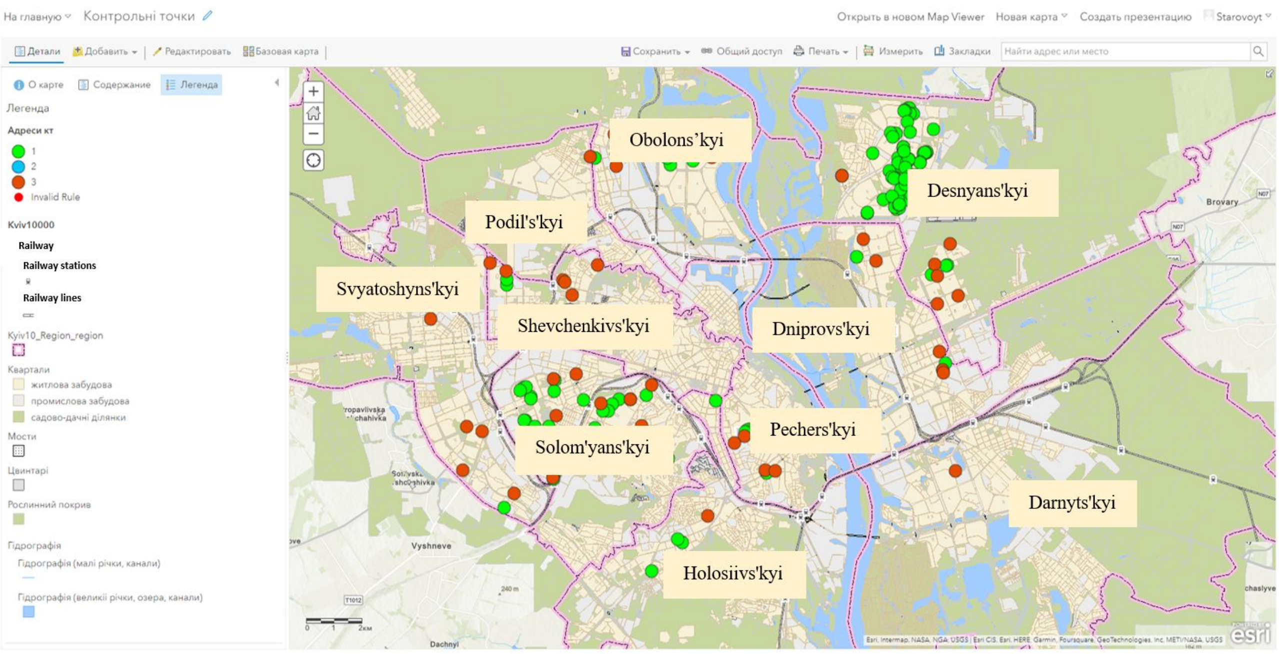

Fig. 8. The result of the transformation of unstructured text information into spatial information

Conclusions

The hybrid quantum-enhanced model of artificial intelligence was developed for the task of fast and automatic transformation of large volumes of unstructured text information of the Kyiv Water Canal into the geodatabase of the GIS water supply system. The model was integrated into GIS using ArcGIS Enterprise. By combining the obtained point classes of spatial objects with already existing data, methods of spatial connections, we developed an interactive map with an update interval of every five minutes. The following conclusions can be drawn from the forecasting model described above:

- Results of performance evaluation and verification of selected indicators: evaluation of accuracy and memorization of models; weighted harmonic mean of precision and recall scores; the Matthews correlation scores showed that the quantum-enhanced QLSTM model and LSTM hybridization with DisCoCat charts did not outperform the classical LSTM model.
- One potential advantage of QNLP is its ability to learn more efficiently and quickly than classical NLP algorithms. Quantum computing is well-suited to processing massive data sets due to its ability to perform many calculations simultaneously.
- The combination of quantum natural language processing and geoinformatics makes it possible to quickly and automatically create geospatial data from unstructured textual information.

Taking advantage of quantum computing and combining it with classical hardware and classical AI models have achieved similar and even better performance in various tasks compared to state-of-the-art methods. Quantum natural language processing is a promising new field that has the potential to revolutionize the way we analyze and understand human language.

Contribution of the authors. Tania Starovoyt – implementation of the methods and empirical data collection and analysis. Yuriy Zaychenko – selection of the methods and methodology of the research, performing literature review, results description and drawing conclusion.

References

Airola, A., Pohjankukka, J., Torppa, J., Middleton, M., Nykänen, V., Heikkonen, J., & Pahikkala, T. (2019). The spatial leave-pair-out cross-validation method for reliable AUC estimation of spatial classifiers. *Data Mining and Knowledge Discovery*, 33(3), 730–747.

Andrade, F. G., Carvalho-Ramalho, R. E., Firmino, A. A., Souza-Baptista, C., Ramos-Falcao, A. G., & Oliveira, M. G. (2020). Using Natural Language Processing for Extracting GeoSpatial Urban Issues Complaints from TV News. *GEOProcessing 2020: The Twelfth International Conference on Advanced Geographic Information Systems, Applications, and Services*. International Academy, Research, and Industry Association.

ArcGIS 2023. https://developers.arcgis.com/python/guide/how-named-entity-recognition-works/

ArcGIS API for Python. https://developers.arcgis.com/python/guide/how-named-entity-recognition-works/

Baidu. *Geocoding API v2.0*. https://api.map.baidu.com/lbsapi/cloud/webservice-geocoding.htm

Bhagvan, K. (2020). *Quantum Computing Solutions: Solving Real-World Problems Using Quantum Computing and Algorithms*. Apress.

Chen, S. (2022). Quantum long short-term memory. *arXiv:2009.01783.* doi:10.48550/arXiv.2009.01783

De Bruijn, J. A., de Moel, H., Jongman, B., de Ruiter, M. C., Wagemaker, J., & Aerts, J.C. (2019). *A global database of historic and real-time flood events based on social media*. Sci. Data.

DisCoCat lambeq. https://cqcl.github.io/lambeq/tutorials/discocat.html

Doccano. https://github.com/doccano/doccano

Enkhsaikhan, M., Holden, E.-J., Duuring, P., & Liu, W. (2021). *Understanding ore-forming conditions using machine reading of text*. Ore Geology Reviews.

Enkhsaikhan, M., Liu, W., Holden, E.-J., & Duuring, P. (2021). Auto-labelling entities in low-resource text: A geological case study. *Knowledge and Information Systems*, 63(3), 695–715.

Floridi, L., & Chiriatti, M. (2020). GPT-3: Its Nature, scope, limits, and consequences. *Minds and Machines*, 30(4), 681–694.

Gonzlez-Castillo, S., & Combarro, E., F. (2023). *A Practical Guide to Quantum Machine Learning and Quantum Optimization*. Packt Publishing.

Guarasci, R., De Pietro, G., & Esposito, M. (2022). *Quantum natural language processing: Challenges and opportunities*. Applied Sciences.

Hoffmann, T. (2021). Quantum Models for WordSense Disambiguation. *Master's thesis in Complex Adaptive Systems.* Chalmers University of Technology.

Hu, Y., Mao, H., & McKenzie, G. (2019). A natural language processing and geospatial clustering framework for harvesting local place names from geotagged housing advertisements. *Int. J. Geogr. Inf. Sci.,* (33), 714–738.

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv:1508.01991v1*. doi:10.48550/arXiv.1508.01991

Jacquier, A., Kondratyev, O., Lipton, A., & López de Prado, M. (2022). *Quantum Machine Learning and Optimisation in Finance*. Packt Publishing.
Lai, J., Lansley, G., Haworth, J., & Cheng, T. (2020). A name-led approach to profile urban places based on geotagged Twitter data. *Trans. GIS 2020,* (24), 858–879.
Lawley, C., Gadd, M. G., Parsa, M., Lederer, G. W., Graham, G. E. & Ford, A. (2023). *Applications of Natural Language Processing to Geoscience Text Data and Prospectivity Modeling*. Natural Resources Research. Springer.
Ma, K., Tian, M., Tan, Y., Xie, X., & Qiu, Q. (2021). *What is this article about? Generative summarization with the BERT model in the geosciences domain*. Earth Science Informatics.
McKenzie, G., Liu, Z., Hu, Y., & Lee, M. (2018). Identifying urban neighborhood names through user-contributed online property listings. *ISPRS Int. J. Geo-Inf.*, (7), 388.
O'Riordan, L. J., Doyle, M., Baruffa, F., & Kannan, V. (2020). A hybrid classical-quantum workflow for natural language processing. *arXiv:2004.06800*. doi:10.48550/arXiv.2004.06800.
Peixeiro, M. (2022). *Time Series Forecasting in Python*. Manning.
PennyLane (2023). https://pennylane.ai/qml/
Qiskit (2023). https://qiskit.org/ecosystem/machine-learning/tutorials/01_neural_networks.html
Rieffel, E. G., & Polak, W. H. (2011). *Quantum Computing: A Gentle Introduction*. MIT Press: Cambridge, MA, USA.
Scikit-learn. https://scikit-learn.org/stable/index.html
spaCy. *Industrial-Strength Natural Language Processing in Python*. https://spacy.io/
The Stanford Natural Language Processing Group. Stanford Named Entity Recognizer (NER). https://nlp.stanford.edu/software/CRF-NER.shtml
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., N, Kaiser Ł., & Polosukhin, I. (2017). Attention Is All You Need. *31st Conference on Neural Information Processing Systems* (NIPS 2017). Long Beach, CA, USA.
Wang, S., Yan X., Zhu, Y., Song, J., Sun, K., Li, W., Hu, L., Qi, Y., & Xu, H. (2022). New Era for Geo-Parsing to Obtain Actual Locations: A Novel Toponym Correction Method Based on Remote Sensing Images. *Remote Sensing. Special Issue "Intelligent Perception in Urban Spaces from Photogrammetry and Remote Sensing*", 14(19), 4725. https://doi.org/10.3390/rs14194725/
Wu, Y., & Wang, Q. (2019). A Categorical Compositional Distributional Modelling for the Language of Life. *arXiv:1902.09303*. https://doi.org/10.48550/arXiv.1902.0930
Zhu, D., Linke, N.M., Benedetti, M., Landsman, K.A., Nguyen, N.H., Alderete, C.H., Perdomo-Ortiz, A., Korda, N., Garfoot, A., & Brecque, C. (2019). *Training of quantum circuits on a hybrid quantum computer.* Science Advances. doi: 10.1126/sciadv.aaw9918

Отримано редакцією журналу / Received: 07.03.23
Прорецензовано / Revised: 14.03.23
Схвалено до друку / Accepted: 25.03.23

Тетяна СТАРОВОЙТ, студ.
ORCID ID: 0009-0008-6335-7679
e-mail: starovoyt.tania@lll.kpi.ua
Національний університет водного господарства та природокористування, Рівне, Україна

Юрій ЗАЙЧЕНКО, д-р техн. наук, проф.
ORCID ID: 0000-0001-9662-3269
e-mail: zaych@i.com.ua
Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”, Київ, Україна

ГІБРИДНА КВАНТОВО-ВДОСКОНАЛЕНА МОДЕЛЬ ШТУЧНОГО ІНТЕЛЕКТУ В ЗАДАЧІ АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ ТА ШВИДКОГО ПЕРЕТВОРЕННЯ НЕСТРУКТУРОВАНОЇ ТЕКСТОВОЇ ІНФОРМАЦІЇ НА ПРОСТОРОВУ

В с т у п . **Ефективне перетворення великого об'єму неструктурованих текстових даних на просторову інформацію є критично важливим для управління системами розподілу води. Це дозволяє здійснювати конверсію великих наборів текстової інформації, таких як звіти, замовлення, листи й інші документи, на точкові класи просторових об'єктів у географічних інформаційних системах. Для опрацювання цієї проблеми, у новому перспективному підході йдеться про поєднання гібридних квантово-класичних нейронних мереж із геоінформаційними технологіями.**

М е т о д и . **Використано гібридні квантово-вдосконалені нейронні мережі разом із методами ГІС для розпізнавання іменованих сутностей, таких як особисті рахунки з їхніми адресами й геокодуванням, та елементи бухгалтерської документації Київводоканалу. Вказана інформація потім оприлюднюється на геопорталі з використанням платформи ArcGIS Enterprise у реальному часі, що є дуже перспективним для ефективного керування розподіленням води. Характеристики розробленої моделі оцінено за показниками точності, параметрами відкликання та зваженим гармонічним середнім значенням точності та відкликання.**

Р е з у л ь т а т и . **Отримані результати вказують, що розроблена гібридна квантово-класична модель штучного інтелекту може бути успішно застосована до трансформації великих об'ємів неструктурованої текстової інформації на просторову. Модель була інтегрована в ГІС із використанням ArcGIS Enterprise платформи. Суміщаючи отримані точкові класи просторових об'єктів з уже існуючими даними та методами просторових поєднань, автори розробили інтерактивну карту з інтервалом оновлення кожні п'ять хвилин.**

В и с н о в к и . **Використовуючи переваги квантових обчислень і поєднуючи їх із класичним апаратним забезпеченням та класичними моделями штучного інтелекту, стало можливим досягти подібних і навіть кращих характеристик порівняно з існуючими сучасними методами для опрацювання різних завдань. Квантове оброблення природної мови є новим перспективним напрямом, який має потенціал докорінно змінити підхід, за яким аналізується та розуміється мова людини.**

К л ю ч о в і с л о в а . **QNLP, гібридні квантово-класичні нейронні мережі, геокодування, геоаналіз, геоінформаційні системи, ArcGIS, розпізнавання сутностей, неструктурована текстова інформація, варіаційний квантовий запальник, QLSTM, просторові об'єкти.**

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2788-6603