IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 June 2023, accepted 12 July 2023, date of publication 14 July 2023, date of current version 20 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3295572

RESEARCH ARTICLE

A Gated Graph Neural Network With Attention for Text Classification Based on Coupled P Systems

JIAQI ZHANG AND XIYU LIU, (Member, IEEE)
Business School, Shandong Normal University, Jinan 250061, China

Corresponding author: Xiyu Liu (xyliu@sdnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62172262, Grant 61876101, Grant 61802234, and Grant 61806114; in part by the China Postdoctoral Special Funding Project under Grant 2019T120607; in part by the Natural Science Foundation of the Shandong Provincial under Grant ZR2019QF007; in part by the China Postdoctoral Science Foundation Funded Project under Grant 2017M612339 and Grant 2018M642695; and in part by the Youth Fund for Humanities and Social Sciences, Ministry of Education, under Grant 19YJCZH244.

ABSTRACT Text classification is a classical task in natural language processing. Prior traditional text classification methods rely on manually extracted features to a great extent, which are easily influenced by human subjectivity. Some existing text classification methods based on the artificial neural networks sometimes neglect the contextual semantic relationships of discontinuous word sequences, resulting in poor learning results. To alleviate these problems, we propose an attention-based gated graph neural network in the framework of coupled P systems (CPGANN) to automatically extract feature representations of nodes. The gating unit with attention is introduced to aggregate neighbor information to capture context semantic relations, and effectively alleviate the long-term dependence on discontinuous words. In order to obtain more discriminative nodes for classification, the attention mechanism is employed in CPGANN to extract keyword nodes before readout to aggregate subgraph representations. Extensive experiments on four real-world datasets demonstrate that CPGANN outperforms all other state-of-the-art baseline algorithms.

INDEX TERMS Attention mechanism, graph neural network, P systems, text classification.

I. INTRODUCTION

Deep learning has made great progress in some tasks such as image processing [1], [2], [3], [4], speech recognition [5], and semantic understanding [6], [7] by dealing with Euclidean data distributed over a high-dimensional feature space through neural networks. In this process, deep learning methods rely heavily on the features extracted by neural networks. In traditional machine learning methods, features are often extracted by experienced experts. Nevertheless, features are automatically extracted by artificial neural networks in deep learning methods that require less manual intervention by contrast. Consequently, deep learning is becoming more and more popular in practical applications for the past few years.

The associate editor coordinating the review of this manuscript and approving it for publication was Rajeswari Sundararajan.

However, there exists a large amount of complex non-Euclidean data in the real world, such as graph data, manifold data, etc. Among them, graph data is a kind of non-linear data consisting of quite a few objects and relationships, which can present the complex relationships between entities in the real world more accurately. Due to its flexibility, graph data has been applied to node classification [7], [8], link prediction [9], [10], [11], text classification [12], clustering [13], [14], and many other areas. Deep learning methods have been widely utilized in recent years to extract features of non-Euclidean spatial data. As a result, Graph Neural Network (GNN) [15] has attracted more and more attention from researchers as a new research hotspot.

Bruna et al. first applied convolutional neural networks to graphs and proposed the Graph Convolution Network (GCN) [16] by transforming the convolution operator, which reduced the computational complexity of GNN models. Since then, researchers have applied the gating mechanism to GNN

based on GRU [17], and Li et al. [18] proposed Gate Graph Neural Network (GGNN) to process sequence data. GGNN considers not only the relationship between the current node and its previous node but also the connection between multiple neighbor nodes and the current node. Velikovi et al. [19] introduced attention mechanism into GNN and proposed Graph Attention Network (GAT). Its attention mechanism can give higher weight to nodes and edges that are more related to neural network tasks, and improve the effectiveness of training and the accuracy of testing. The gating mechanism and attention mechanism have been also successfully applied to many real problems of GNN nowadays. Wu et al. [20] proposed a fused gated graph convolutional network model for rumor detection research, which can effectively highlight the importance of source post information. In the task of machine translation [21], the importance of different words in the input sentence is reflected by the mechanism of distributing attention weights.

Text classification is one of the most fundamental tasks in natural language processing. The traditional shallow learning model is adopted to deal with the text classification problem mainly by ignoring the grammar and order of some texts and representing a text or paragraph with some specific symbols. Such models have few parameters and good domain adaptability. For instance, the bag of words model [22] employs a set of disordered sequences of words to express a text or a document, puts all the words in the whole document set into the bag, and then arranges them disordered.

In information retrieval and text mining, TF-IDF [23] is a commonly used weighting technique to evaluate the importance of words in a single document or whole corpus, yet only adopting word frequency to measure the importance of words seems to be defective. These shallow learning methods rely on manually acquired text features to a great extent. With the explosive growth of information, manual annotation data has become time-consuming and easily susceptible to the subjective consciousness of people. The other type of deep learning model is more complex in structure, which does not rely on the manually extracted text features, and can directly model the text content. For example, Kim et al. [24] proposed to use CNN for text classification based on word vector representation. Johnson et al. [25] proposed a text classification model based on LSTM adopting supervised learning and semi-supervised pre-training method. Yao et al. [12] constructed a heterogeneous graph based on global relationships between text and word for semi-supervised text classification on GCN. But it did not consider fine-grained text-level word interactions, resulting in insufficient capture of semantic information. When there are more nodes and larger graphs, the full graph update calculates the entire corpus that contains too much redundancy, increasing the time complexity and requiring larger memory space.

Membrane computing was formally proposed in 1998 by Păun [26], also known as P systems. It belongs to a branch of natural computing, which is a biological computational model abstracted according to biological structure and function. In recent years, it has attracted the attention of many scholars because of its characteristics of distributed, non-deterministic, and parallelism. According to different structures and functions, cell-like P system [27], tissue-like P system [28], and neural-like P system [29] are three categories of P systems.

At present, the current research on P systems mainly focuses on theory and application. Theoretically, various computational models are mainly based on the structural characteristics of biological cells and analyze their computational performance. Wu et al. [30] demonstrated unpolarized cell-like P systems with active membranes, which solves the NP-hard problems from a new perspective and significantly improves the computing capability. The notion of collaboration has been presented in tissue-like P systems [31], where the rules were separately invoked during computation. Additionally, the major application of P systems was to use the designed membrane calculation model to solve some practical problems. Wu et al. [32] defined anti-pulse rules in SN P systems where synapses can change pulses into anti-pulses or anti-pulses into pulses. The results showed that the form of pulse rules is sufficient for Turing generality for ASN P systems with synaptic rules. In terms of applications, the designed P systems model was mainly used to solve some practical problems. Bernardini et al. [33] proposed a P systems model of a population-sensing regulatory network of the bacterium Vibrio fischeri, simulating the cancer pathogenesis process with cell-like P systems. Liu et al. [34] combined the simple ability of rough set attributes with the apoptosis mechanism of biological neurons in the P system effectively for the first time, and designed an apoptosis algorithm of conditional neurons to remove error messages.

In the era of big data, the amount of textual data on the network is growing day by day. As the most widely distributed and data-rich information carrier, text contains rich semantic and contextual information. The purpose of text classification is to organize and categorize text resources. At the same time, it is also a key link to solve the text information overload problem. Therefore, it is especially important to adopt an artificial neural network model to organize and manage the massive text data scientifically. Although the text features that can be extracted by artificial neural networks have been more comprehensive, they are all in the case of a single task, and the situation is less effective if the features are extracted under multiple tasks.

Neural network approximation optimization algorithm based on membrane systems is a hot research direction in membrane computing applications at present, but the research is mostly focused on theoretical aspects. Therefore, a novel coupled P system is constructed fusing the structure of cell-like P systems and tissue-like P systems to alleviate the above deficiencies. It is abstracted from the structure of biological cells and tissues, nesting the structure of cells within tissues and adopting hierarchical computational models within the cell to extract features in parallel. Furthermore, the attention-based graph neural network is combined

with the coupled P systems. The proposed model was inspired by biological structures and abstracted a new computational model, while simultaneously taking advantage of the great parallelism computational properties of the P systems as well as the way of graph neural network processes the data. Moreover, the neural network algorithms can run within the framework of this model. In this way, we can employ the contextual semantic relations captured by graph neural networks to attain more precise semantic node features and filter out more meaningful words for text classification tasks. Thus, CPGANN has a great degree of advantage in dealing with multi-task parallelism problems.

The major contributions of this paper are as follows.

1) In this paper, the structures of cells and tissues in biology are integrated to propose a novel coupled P system that nests cell-like P systems within tissue-like P systems. It is combined with an improved graph neural network so that the neural network model can run within the P systems framework to improve the computational effectiveness by exploiting the great parallelism characteristic of P systems.
2) We propose an Attention Gated Feature Extraction (AGFE) module to perform feature extraction separately. Moreover, different numbers of neighboring nodes' information can be integrated by stacking multiple AGFE layers. For the sake of obtaining a more accurate representation of the subgraph and reducing the computational effort, an attention mechanism before readout is introduced to extract keyword nodes that are more discriminating for the classification task.
3) Experiments are conducted on four real-world datasets to validate the effectiveness of CPGANN, and the results prove that our method outperforms all other baselines.
4) The newly proposed coupled P systems was abstracted from real cellular communication mechanism, so it has practical significance in biology. Since the information transmission mechanism in neural networks is similar to the coupled P systems, CPGANN is an innovative combination that can simultaneously take advantage of both to solve multi-task parallel classification task problems.

The remainder of the article is distributed as follows. In Section II, the related work of this paper will be introduced, and the cell-like P systems and tissue-like P systems will be defined specifically in this part. Section III details the process of CPGANN. The experimental specifics and analysis of results will be demonstrated in Section IV. Finally, Section V summarizes the paper and gives a prospection of future works.

II. RELATED WORK

A. CELL-LIKE P SYSTEMS

The cell-like P systems composed of membrane structure, objects, and rules was the first to be proposed among all P systems [27]. It mimics the structure and function of a cell in biology, whose inner region is split into different small regions by membrane. Fig.1 demonstrates the fundamental structure of cell-like P systems. The exosphere membrane called the skin membrane, isolates cells from their surroundings. The elementary membrane refers to the one that doesn't contain other membranes inside, otherwise, they are non-elementary membranes.

The formal definition of the cell-like P systems is as follows:

$$\Pi = (O, H, \mu, w_i, R_i, i_{0ut}), \tag{1}$$

where

1) O represents the alphabet with elements containing all the objects;
2) H is the collection of membrane labels;
3) μ indicates the membrane structure;
4) w_i means the multiple set of objects;
5) R_i refers to the finite set of rules contained within the membrane;
6) i_{0ut} stands for the output cell.

B. TISSUE-LIKE P SYSTEMS

The tissue-like P systems are proposed based on the way of object transmission and communication by cell group structure in tissues [28]. The basic structure of the tissue-like P systems is expressed in Fig.2 Cell 0 is the input cell, containing initial objects. Cell n is the output cell, storing calculation results. Other cells execute evolutionary rules for message transmission. The rules are triggered only when the required trigger conditions for the rules are met.

The following is the formal description of the tissue-like P systems.

$$\Pi = (O, \sigma_1, \cdots, \sigma_n, syn, i_{out}), \tag{2}$$

where

1) O represents the alphabet with elements containing all the objects;
2) $\sigma_1, \cdots, \sigma_n$ means n cells in the P systems, which is defined as follows:

$$\sigma_i = \left(Q_i, s_{i,0}, w_{i,0}, P_i\right), \quad 1 \le i \le n$$

where

Q_i	stands for finite state set;
$s_{i,0} \in Q_i$	indicate the initial state;
$w_{i,0} \in O^*$	refers to the multiset of initial objects;
P_i	represents a finite set of all rules of the P systems;

3) $syn \subseteq \{0, 1, \cdots, n\} \times \{0, 1, \cdots, n\}$ refers to the synapses between cells to transform the information;
4) i_{out} represents the output cell.

C. PRELIMINARY PREPARATIONS

Given an original document, we first need to construct the document into a graph structure for subsequent operations. Each document generates a graph, with each unique word

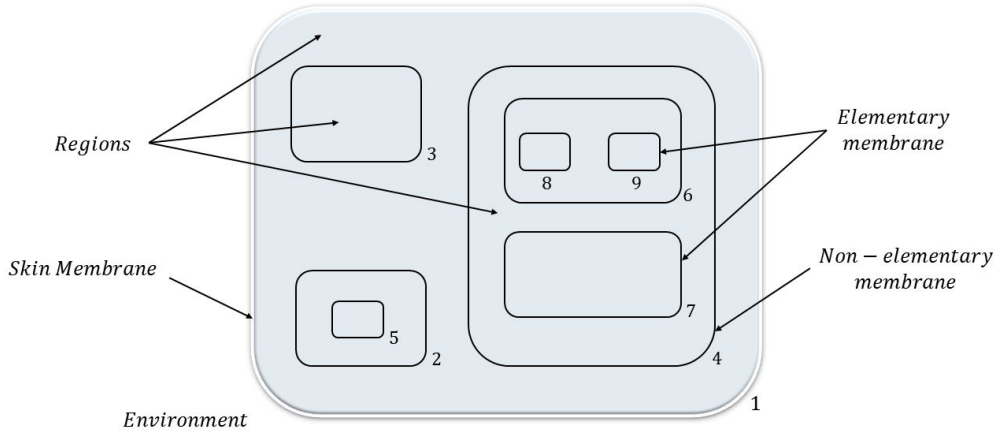


FIGURE 1. The basic structure of the cell-like P systems.

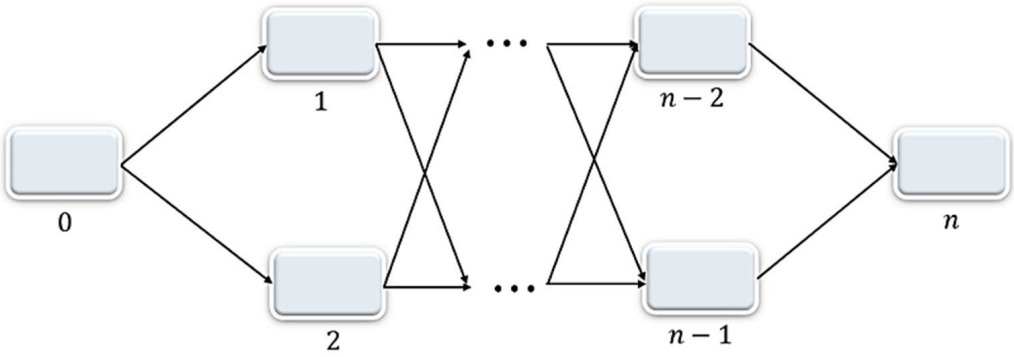


FIGURE 2. The basic structure of the tissue-like P systems.

as a vertex and co-occurrence relationships between words as edges. There is a co-occurrence relationship if the words appear at the same time within an invariant size of the sliding window, i.e., there is an edge between two vertices. Otherwise, there is no co-occurrence relationship, i.e., no edge between two vertices. In this article, the sliding window size defaults to 3. We define the constructed graph as $G = (V, E)$. V is the collection of vertices and E is the collection of edges between vertices. The adjacency matrix of graph G is defined as A. $A_{ij} = 1$ if there is an edge $e_{ij} \epsilon E$ between nodes v_i and v_j, otherwise $A_{ij} = 0$. D is the diagonal matrix of graph G, where $D_{ii} = \Sigma_j A_{ij}$. $\hat{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ is the symmetric normalized adjacency matrix.

In addition, we need to adopt some traditional methods to preprocess the text before we input it into the model for training. These methods include stopwords removal [35], tokenization [36] and so on, which will save memory space and increase computational efficiency. The feature representation h of the word is defined as the initialization input of the node embedding, which is initialized by the feature matrix X of the word. Then, after input h to the attention-gating component in corresponding cells, the whole coupled P system begins to propagate the information. The neighbor features are aggregated to extract the features of each central node.

III. METHOD

The overall framework of the improved gated graph neural network is expressed in Fig.3. The whole process consists of three steps which are Construct Graph Structure, Attention Gating Feature Extraction (AGFE), and Readout. The details of each component will be described in subsequent content.

A. IMPROVED GATED GRAPH NEURAL NETWORK BASED ON COUPLED P SYSTEMS

The coupled P systems are abstracted from biological mechanisms that integrate the structures of tissues and cells, which further nest the cell-like P systems within the tissue-like P systems. In this paper, we put the improved graph neural network model process into the framework of the coupled P systems to run. On the one hand, the characteristic of independent computation without interfering with each other in the cell-like P systems can be exploited. On the other hand, the mechanism of information propagation and communication between cells through synapses in tissue-like P systems

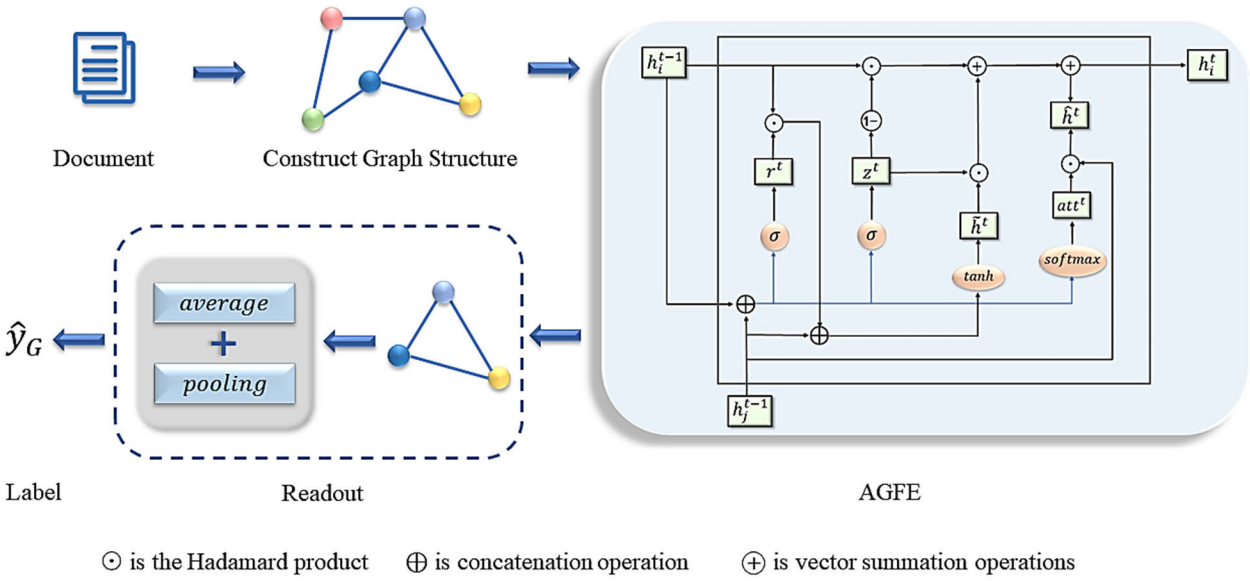


FIGURE 3. The overall framework of improved gated graph neural network.

can be adopted. The coupled P systems define computation methods within cells as evolutionary rules, whereupon we accordingly abstract the corresponding computations in graph neural networks to these rules. The intracellular objects are then operated according to the evolutionary rules, and the obtained results are input to the next cell through the intercellular synapses. The communication rules exist on the synapses for intercellular information transfer and interaction. Ultimately, P systems output the results of the desired classification task. The overall structure of the coupled P systems is illustrated in Fig.4.

The formal definition of the coupled P systems is as follows:

$$\Pi = (O, \eta, \mu, syn, \sigma, R, in, out), \tag{3}$$

where

1) O represents the alphabet with elements containing all the objects;
2) $\eta = \{G, h_1^1, h_2^1, \ldots, h_n^1, A, W, U, b, y_i\} \in O$ indicates all the initial objects;
3) μ stands for the membrane structure;
4) $syn = \{(0,1), (1,2), (2,3), (3,4), (0,4)\}$ describes the synapses that connect cells;
5) σ indicates the cells in the P systems, where $\sigma_1, \sigma_2, \sigma_3$ means the coupled cells, $\sigma_1^1, \ldots, \sigma_1^n$, $\sigma_2^1, \ldots, \sigma_2^n$, $\sigma_3^1, \ldots, \sigma_3^k$ represent the inner layer cells in the coupled cells, σ_4 and σ_5 represent the normal cells;
6) R refers to all the rules in coupled P systems, including evolution rules and communication rules;
7) in is Cell 0, the input cell; out is Cell 4, the output cell.

It can be clearly seen from Fig.4 that there are multiple cells contained in the coupled P systems, where within the outer cells σ_1, σ_2, and σ_3, there are also multiple inner cells σ_1^i, σ_2^i, and σ_3^i nested. σ_1, σ_2 and σ_3 operate independently of each other according to the rules on their respective intercellular synapses. These cells can then be triggered and computed simultaneously by the rules. In addition, the inner layer cells are also independent of each other, and the operations do not interfere with each other. Consequently, they do not have to wait for the end of the previous cell operation before

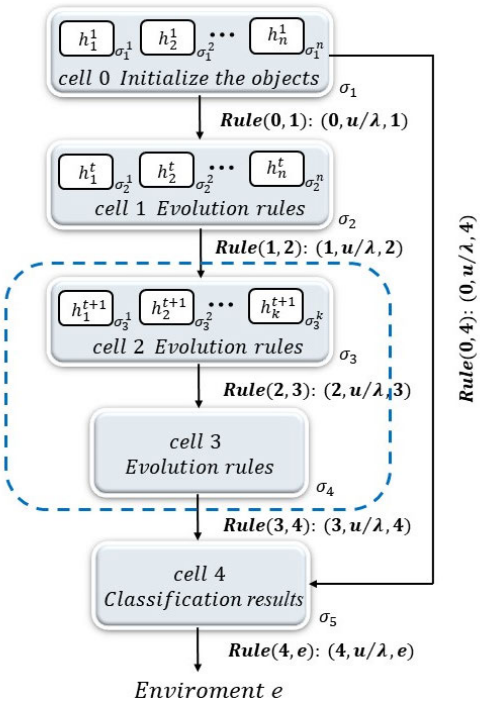


FIGURE 4. The overall structure of the coupled P systems.

turning on the computation. In view of the above computing mechanism, the coupled P systems can take advantage of its unique parallelism feature to theoretically improve the computing efficiency of the model and reduce the computing time.

B. THE EVOLUTION RULES IN DIFFERENT CELLS

The Cell 0 contains all the initialization objects of the P systems and some parameter information. $\eta = \{G, h_1^1, h_2^1, \ldots, h_n^1, A, W, U, b, y_i\} \in O$, which consists of the initial topology graph G constructed from the document, the initialization node embeddings $h_1^1, h_2^1, \ldots, h_n^1$, the adjacency matrix A and the trainable parameters W, U, b. They will all be passed to the subsequent Cell 1 by the communication rule $Rule(0, 1)$. The ground-truth node label y_i in the initial graph will be transferred to Cell 4 by the communication rule $Rule(0, 4)$. The inner layer cells in Cell 0, Cell 1, and Cell 2 are isolated without interfering with each other. They independently and in parallel execute evolutionary rules in their respective cells. Furthermore, Cell 0, Cell 1, Cell 2, Cell 3, and Cell 4 also execute the evolutionary rules in their respective cells independently without interfering with each other. The information can be disseminated and interacted mutually at any time by rules in the coupled P systems.

1) THE EVOLUTION RULES OF AGFE IN CELL 1

An attention-gated feature extraction operation will be performed in Cell 1. We make use of an attention mechanism in the gated graph neural network to aggregate the information between the central node and its first-order neighbors, which is more effective in capturing contextual information and enhancing the semantic information propagation process. In this part, different weights are assigned depending on the degree of contribution of the neighbors to the central node. The h_i^t in Cell 1 denotes the node representation of the i-th inner cell at t moment. R_{11} calculates the attention score of each neighbor node and then normalizes it by *softmax* function to obtain att_j^t, which indicates the importance degree of neighbor node j to the central node i.

$$R_{11} : att_j^t = softmax\left(h_j^{t-1}\right) = \frac{exp\left(h_i^{t-1}W\right)}{\sum_{j\in N} exp\left(h_j^{t-1}W\right)}, \quad (4)$$

where h_j^{t-1} indicates the node representation of the first-order neighbor j of node i at t-1 moment. N refers to the set of all 1-hop neighbors of node i. W refers to the shared linear transformation matrix. After that, we compute the attention-weighted representation of each node by R_{12}.

$$R_{12} : \hat{h}_i^t = \sum_{j\in N} att_j^t h_j^{t-1}, \quad (5)$$

where N still refers to the set of 1-hop neighbors of node i. By stacking t layers AGFE, we can aggregate the feature information of t-hop neighbors. In the t-th AGFE layer, $R_{13}, R_{14}, R_{15}, R_{16}$ describe the evolution rules of the gating unit.

$$R_{13} : x^t = Ah^{t-1}W_x, \quad (6)$$

$$R_{14} : z^t = \sigma\left(W_z x^t + U_z h^{t-1} + b_z\right), \quad (7)$$

$$R_{15} : r^t = \sigma\left(W_r x^t + U_r h^{t-1} + b_r\right), \quad (8)$$

$$R_{16} : \tilde{h}^t = tanh\left(W_h x^t + U_h\left(r^t \odot h^{t-1}\right) + b_h\right). \quad (9)$$

The x^t obtained by R_{13} expresses the information aggregated at t moment from the neighbors. z^t in R_{14} is the update gate which integrates the node representation of the previous moment and the massage of neighbors, deciding how much information will be remembered to pass on to the next unit. r^t in R_{15} is the reset gate which determines how much information will be forgotten, filtering out excess redundant massage in order to pass the more important information to the next unit. σ performs the sigmoid function. W and U are the trainable weight, b means the bias. $\tilde{h}^t$ denotes the newly updated hidden states calculated using the *tanh* activation function used to store intermediate information. Eventually, we can obtain the node embedding matrix h^t for layer t by R_{17}.

$$R_{17} : h^t = \tilde{h}^t \odot z^t + h^{t-1} \odot \left(1 - z^t\right) + \hat{h}^t, \quad (10)$$

where $\odot$ is the Hadamard product. A more intuitive computational flow is demonstrated in the AGFE module in Fig.3.

2) THE EVOLUTION RULES OF EXTRACTING KEYWORDS IN CELL 2

In Cell 1, we have computed the embedding representation of each node in graph G. For a document, the category into which it is classified is often determined by a few keywords to a great extent. Sometimes not every word in the document is necessarily useful for the classification task, which may instead cause data redundancy and increase computational effort. It is quite important to extract the discriminable words that contribute more to the text classification. Therefore, CPGANN applied an attention mechanism and pooling operation to extract keyword nodes in the P systems.

$$R_{21} : q^t = Att\left(h^t\right), \quad (11)$$

R_{21} obtains the attentional score q^t for each node, and Att (.) is defined as the attention function. h^t is hidden state of the node at current moment.

$$R_{22} : M^t = Rank\left(q^t, k\right), \quad (12)$$

Then we rank the attention scores of each node obtained by R_{21} from highest to lowest. R_{22} selects the nodes whose attention score is top k by the *Rank* function, and the set of k nodes is M^t.

$$R_{23} : h'^t = q^t \odot h^t, \quad (13)$$

R_{23} updated attention-weighted hidden state h'^t by attention score q^t and current hidden state h^t.

$$R_{24} : h^{t+1} = A'^t h'^t, \quad (14)$$

The state of the next moment h^{t+1} is calculated by R_{24}, where $A_i'^t = \hat{A}_i^t$, $i \in M^t$. Thus, R_{24} can iteratively update the hidden states of nodes in Cell 2.

3) THE EVOLUTION RULES OF READOUT IN CELL 3

In Cell 2, we have attained the updated node representations and a collection of nodes consisting of keywords. However, as far as the classification task be concerned, it is far from enough to just obtain a node-level representation. We need to aggregate these nodes and readout the node-level representation to generate a graph-level representation in Cell 3. The keyword subgraph needs to be constructed by R_3.

$$R_3 : h_G = \frac{1}{k}\sum_{v=1}^{k} h_v^{t+1} + pooling\left(h_1, \cdots, h_k\right), \tag{15}$$

R_3 contains an *average* operation and a *maxpooling* operation to aggregate node features with important neighboring node features. On the basis of averaging all nodes in the subgraph, the keyword information in the text is captured by employing *maxpooling*. These operations allow the model to preserve original features while reducing the parameters of network training, making the training time decrease. In this way, we can acquire a more effective subgraph representation result.

4) THE EVOLUTION RULES OF CLASSIFICATION IN CELL 4

The classification of text will be performed in Cell 4 and the cross-entropy objective function will be optimized.

$$R_{41} : \hat{y}_G = softmax\left(ReLU\left(Wh_G + b\right)\right), \tag{16}$$

$$R_{42} : \mathcal{L} = -\sum_i y_i log\left(\hat{y}_i\right), \tag{17}$$

where $\hat{y}_G$ is the prediction label of the output in R_{41}, $ReLU$, and $softmax$ are activation functions. W is the matrix of a linear transformation that maps the embedding representation to the desired output space. b is the bias term. While in R_{42}, y_i is the i-th element of the one-hot ground truth label of graph G, $\hat{y}_i$ is the i-th element of the predicted label of graph G, $\mathcal{L}$ is devised to calculate the cross-entropy loss. We aim to minimize the distance between these two labels.

C. THE COMMUNICATION RULES BETWEEN DIFFERENT CELLS

In the coupled P systems, cells transmit and communicate information through synapses between each other, which means that information is only exchanged between different cells when synapses are present. We define certain communication rules on the synapse and the rules are directed. The cells can then communicate with each other in the order we want. In general, rules are divided into unidirectional transmission and bidirectional transmission, and in this paper only rules exist for unidirectional transmission from cells to other cells or from cells to the environment.

Definition of Communication Rules: Rule $(i, j) : (i, u/\lambda, j)$ denotes the communication rule from Cell i to Cell j. u refers to the information passed from Cell i to Cell j, and λ represents the information transmitted from Cell j to Cell i. Since the rules in this paper are all unidirectional rules, no information is transmitted from Cell j to Cell i, i.e., λ is null.

The communication rules are specified in this paper as:

Rule$(0, 1) : (0, u/\lambda, 1)$ passes the object η from Cell 0 to Cell 1, which contains initial topology graph G constructed from the document, the initialization node embedding representations $h_1^1, h_2^1, \ldots, h_n^1$, the adjacency matrix A and the trainable parameters W, U, b;

Rule$(0, 4) : (0, u/\lambda, 4)$ transfers the ground truth label of every node from Cell 0 to Cell 4 for the purpose of calculating the distance between the true label y_i and predicted label $\hat{y}_i$ in Cell 4 and optimizing the objective function $\mathcal{L}$.

Rule$(1, 2) : (1, u/\lambda, 2)$ inputs the feature embedding representation of each node calculated in AGFE module of Cell 1 into Cell 2. This rule conveys the same amount of node information as *Rule*$(0, 1) : (0, u/\lambda, 1)$, both for n nodes.

Rule$(2, 3) : (2, u/\lambda, 3)$ transfers keyword set M^t including k keyword nodes and the updated current hidden state h^{t+1} extracted from Cell 2 into Cell 3 to aggregate a graph-level representation. Since the node-level representation is not applicable to the text classification task, what we want to do is to classify the graph G constructed from the text document;

Rule$(3, 4) : (3, u/\lambda, 4)$ delivers the subgraph representation h_G obtained from the Readout in Cell 3 to Cell 4, which is composed of k keywords that contribute most to the text classification task.

Rule$(4, e) : (4, u/\lambda, e)$ transmits the result of text classification and the loss of the whole model calculated by coupled P systems in Cell 4 to the environment, and one iteration ends.

IV. EXPERIMENTS

A. DATASETS

In the experimental part of this paper, four datasets were chosen to verify the classification performance of the model.

SST2[1] (The Stanford Sentiment Treebank) is a dataset of movie reviews, each sample contains a sentence and the corresponding label of the sentence. It is divided into 2 categories, positive sentiment (sample label corresponds to 1) and negative sentiment (sample label corresponds to 0). Of the total 9613 documents, 7792 documents are utilized in training and 1821 documents are used for testing.

R8 and R52[2] are two subsets of Reuters News. There were 7674 documents in R8, divided into 8 categories, of which 2189 documents are employed for testing and 5485 documents are employed for training. There are 9100 documents in R52, divided into 52 classes. 6532 training documents and 2568 test documents are included.

Ohsumed[3] is derived from MEDLINE10, a database of pharmaceutical information, which contains titles and

[1] http://nlp.stanford.edu/sentiment/
[2] https:// www. cs. umb. edu/ ∼smima rog/ textm ining/ datas ets/
[3] http:// disi. unitn. it/ mosch itti/ corpo ra. htm

TABLE 1. Statistics of datasets.

Datasets	Docs	Train	Test	Classes	Avg.length
SST2	9613	7792	1821	2	35
R8	7674	5485	2189	8	65
R52	9100	6532	2568	52	69
Ohsumed	7400	3357	4043	23	136

abstracts from 270 pharmaceutical journals over five years. The total of 7400 documents contains medical abstracts for 23 cardiovascular disease categories. 3357 documents are selected for training and 4043 documents are selected for testing.

TABLE 1 shows the statistic details of the datasets. As we can see in IV-B, Docs is the number of documents contained in the dataset, Train and Test denote the number of documents used for training and testing, respectively. Avg.length means the average length of sentences in the documents. Among all the datasets, SST2 and R8 are short text datasets, while R52 and Ohsumed are long text datasets.

B. BASELINES

In this paper, we compare CPGANN with the following baselines to verify their effectiveness.

LSTM [37]: This is a multi-task shared RNN framework, which aims to solve the long-term dependence of general RNN. It exploits multiple different task datasets to train the same model shared parameters, and has the effect of expanding the dataset.

CNN [24]: Convolutional Neural Networks is applied to the sentence classification task via a pre-trained word vector and improves the performance by learning the task-specific word vectors.

SWEM [38]: The model performs simple pooling operations on the word vector, and systematically compares the simple method with pooling based on word embedding with the complex model such as LSTM and CNN.

Transformer [39]: This is a model that does not use convolutional or recurrent network layers, while is based entirely on an attention mechanism that combines self-attention and multi-head attention.

Text-Level-GNN [40]: Instead of constructing a single graph for the whole corpus, this model constructs a global parameter sharing graph for each input text. Text-Level-GNN eliminates the burden of dependencies between a single text and the whole corpus, supporting online tests but still retaining global information.

Text-GCN [12]: This is a model of GCN model for text classification tasks. The model builds graphs of text in the corpus based on word co-occurrence relationship between words in original text, and then GCN learns the vector representation of nodes for text classification.

FastText [41]: It is a simple but effective method of text classification. By introducing other statistics, it reduces the accuracy gap between the linear and deep models. The model has only one hidden layer and one output layer, thus greatly reducing the training time.

HGAT [42]: This is a new semi-supervised method based on the heterogeneous graph neural network for short text classification. HGAT utilizes the heterogeneous graph convolution to account for the heterogeneity of different information types and realizes the semi-supervised short text classification through the propagation of information along graphs.

BERT [43]: BERT is no longer only focusing on the information before or after a word, but on the context information of the entire layers of its entire model, pre-training the deep two-way representation by joint context adjustment in all layers.

TM [44]: Tsetlin Machine (TM) is a pattern recognition method that applies to propositional formulas. It extracts semantically related words from pre-trained word representations and learns features with additional correlations based on comparing pre-trained word representations using cosine similarity. In this way, TM can improve the model performance while maintaining the model interpretability.

UGformer [45]: Transformer is applied in GNN to extract the information and learn the representations of the graph in this paper. Two model variants are designed: one is to use Transformer on the set of sampled neighbors of each input node, and the other is to use Transformer on all the input nodes.

Text-MGNN [46]: Text-MGNN is a novel GNN from the perspective of multi-granular topic perception. Topic nodes are introduced to construct a triple node set and build a multi-granular relationship model on the text graph of this triple node set. This method not only enhances the dissemination of information but also reduces the impact of heterogeneous information caused by polysemous words.

C. RESULTS AND ANALYSIS

The validity of CPGANN and baselines was evaluated with accuracy and F1-score in this paper. There are many parameters in this paper, where the learning rate was set to 0.01, the number of iterations was 200, the dropout rate was 0.5, and the sliding window size was fixed to 3. The number of layers was set to two layers, that is, two AGFE layers were stacked to capture the first and second order neighbors' feature information. All experiments were performed on a PC with Windows10 64-bit operating system, 2.4 GHz CPU, 16 GB MAP, and Intel Core i5-1135G7, running the code with python3.7.

Experimental results are demonstrated in TABLE 2, and some of the baseline experimental results were obtained from [44], [45], [46], [47], [48], and [49]. It is evident to find that CPGANN outperforms other baselines in the vast majority of cases. Whether on the two short document datasets, SST2 and R8, or the two long document datasets, R52 and Ohsumed, the results of CPGANN are the most optimal. It is clear in TABLE 2 that these methods generally perform better on R8 and R52, and slightly worse on the long text

TABLE 2. Test accuracy(%) of our method against other baselines on four datasets. The results were reported with mean ± standard deviation.

Model	SST2	R8	R52	Ohsumed
LSTM	80.41±0.20	96.09±0.19	90.48±0.86	51.10±1.50
CNN	80.47±0.58	95.71±0.52	87.59±0.48	58.44±1.06
SWEM	63.12±0.55	95.32±0.26	92.94±0.24	63.12±0.55
Trans	76.93±0.39	87.33±0.23	76.98±0.07	66.72±0.58
Text-Level-GNN	81.47±0.26	97.80±0.20	93.60±0.30	69.04±0.60
TextGCN	82.36±0.11	97.07±0.10	93.56±0.18	68.36±0.56
FastText	81.93±0.66	96.13±0.21	92.81±0.09	57.70±0.49
HGAT	81.68±0.14	93.74±0.10	92.75±0.22	52.68±0.18
Bert	80.78±0.42	96.67±0.22	90.38±0.10	62.75±0.26
TM	77.51±0.60	97.50±1.12	88.59±1.20	-
UGformer	79.86±0.29	97.05±0.32	94.71±0.42	**70.63±0.18**
Text-MGNN	75.54±0.22	97.39±0.74	94.20±0.13	70.00±0.36
CPGANN	**84.67±0.41**	**97.97±0.28**	**95.12±0.69**	69.61±0.21

dataset Ohsumed. What's more, the results of baselines on R52 were not as good on R8, because R52 has far more classes than R8 and a longer training time. Both TextGCN and Text-Level-GNN build a graph for each document rather than train based on the whole corpus, so this kind of graph neural network algorithm performs well on all datasets. However, it is worth noting that since TextGCN updates full graph when information iterates, it means that its computation time is slow and memory consumption is large if there are more nodes and larger graphs. Relatively speaking, the effect of CPGANN is improved to some extent by using attention mechanism. Owing to SWEM models for the word vector, the document-level word frequency information and word co-occurrence information is insufficient. Besides, SWEM does not consider the influence of word order factors. As a consequence, it performs worst on SST2. HGAT takes the heterogeneity of different node types into account. Its two-level attention mechanism can obtain different importance coefficients of different neighboring nodes. Nevertheless, HGAT is poor in the processing of long document data and is more suitable for short text data. Therefore, its results on SST2 and R8 are better than R52 and Ohsumed. Since Transformer introduces the attention mechanism, the performance is improved compared to SWEM, but it focuses more on the global correlation. The local information acquisition ability is not as strong as LSTM and CNN, so the results are worse than theirs. The performance of Bert turns out to be relatively great due to it paying attention to the context information, not only aggregating the information of the words before and after. Though Bert has a stronger semantic understanding capability, the processing of long text data needs to be strengthened. FastText averages the word and n-gram vectors over the whole document to obtain the document vector and then utilizes the document vector to perform softmax multi-classification. The structure of FastText compared with CPGANN is too simple to capture the word order features, so its capability is less effective than CPGANN. It is also clear from TABLE 2 that the results of TM are much worse than that of CPGANN. Since UGformer applies transformer to GNN, it has more advantages in the process of representing graphs. In addition, it also introduces a self-attention mechanism, so the performance of UGformer is better than Transformer. In particular, it performs best on the Ohsumed dataset, even exceeding CPGANN. The triple node set introduced by Text-MGNN can strengthen the semantic propagation process of words in documents, and further promote the representation learning of graphs. The performance of this method is worse than that of CPGANN on SST2, R8, and R52 datasets. On the other hand, CPGANN is inferior to Text-MGNN because of its shortcomings in handling long text datasets.

In summary, CPGANN consistently outperforms other baseline algorithms in general. The above results verify that CPGANN does have an outstanding ability to process and classify text information. Fig.5 expresses the accuracy comparison on different models, which can more intuitively indicate the improvement degree of CPGANN.

D. PARAMETER SENSITIVITY

In this section, we will analyze the impact of different parameter settings on the model performance in terms of sliding window size, learning rate, the number of AGFE layers, and dropout on SST2, Ohsumed, R8, and R52 datasets. Experiments were carried out on four datasets in the first section, and only sliding window or learning rate were changed respectively to verify the impact on model performance. The second section takes SST2 as an example, discussing the influence of two parameters changing at the same time. We combine the four parameters mentioned above in pairs, and there are 6 kinds of ways of permutation. All other parameter settings are identical except for the parameter to be analyzed.

1) THE EFFECT OF SINGLE PARAMETER

a: SLIDING WINDOW SIZE

Word co-occurrence information is counted using a fixed-size sliding window sliding through the corpus. If two words

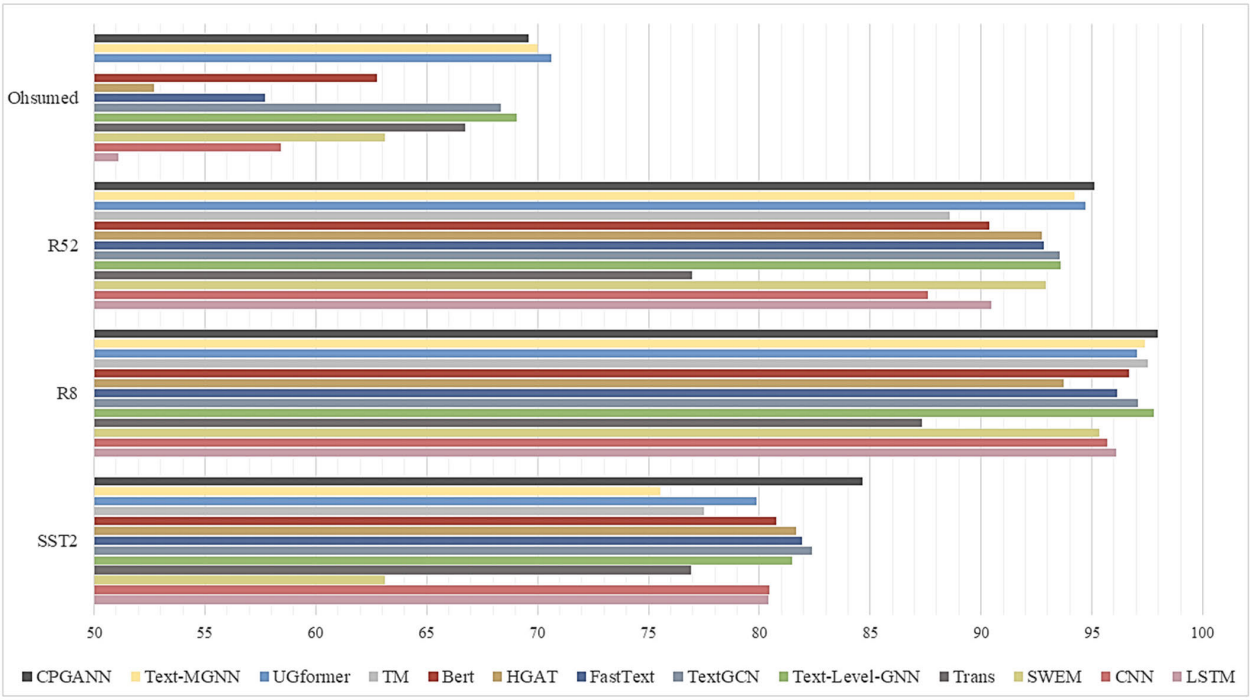


FIGURE 5. Comparison of accuracy(%) between different methods on four datasets.

appear in the same window all at once, indicating that there is a co-occurrence relationship between two words and an edge is added between them. A suitable size of the sliding window is significant for constructing the graph structure of the documents. We conducted experiments on each of the four datasets to verify which window size is best for CPGANN. The F1-score is illustrated in Fig.6.

When the sliding window size is 3, the F1-score significantly improves on four datasets, indicating that the model performs best at this time. When the window size continues to increase, the F1-score evidently shows a decreasing trend for both Macro-F1 and Micro-F1. Finally, we choose a sliding window size of 3.

b: LEARNING RATE

We perform experiments to explore the learning rate that best fits this model, and the obtained F1-score is expressed in Fig.7. As is illustrated that the model performance starts to improve with the increase of the learning rate from 0.001, and decreases after reaching a certain value. Most of the datasets have optimal results when the learning rate reaches 0.01, F1-score starts to decrease as the learning rate continues to increase. The trend of the results on SST2 and R8 is not particularly obvious, while R52 and Ohsumed are relatively sensitive to the change of learning rate. Eventually, we set the learning rate to 0.01.

2) THE EFFECT OF TWO PARAMETERS

We take SST2 as an example to explore the influence of any two parameter changes on the experimental accuracy. The results are similar for other datasets. As shown in Fig.8(a), it exhibits the influence on classification accuracy under the combination of sliding window and learning rate. The result is slightly better when the sliding window is 2 and the learning rate is 0.01. Other else changed little. The effects of simultaneous changes in sliding window and dropout are expressed in Fig.8(b). The overall fluctuation is also insignificant. Fig.8(c) and (d) indicate the effects of the changes of the two pairs of parameters which include sliding windows and AGFE layer, learning rate, and AGFE layer, respectively. When the number of AGFE layers is fixed, the variation brought by dropout appears to be larger than that brought by the sliding window size. In addition, Fig.8(e) exhibits the impact of changes in learning rate and dropout. Fig.8(f) displays the changes caused by dropout and AGFE layer. Different from several other plots, (e) and(f) have more significant changes when the learning rate is 0.1, especially when combined with the AGFE layer, the results are more volatile. This phenomenon manifests that learning rate and the number of AGFE layers have a greater effect on the model.

In conclusion, the performance of the model does not change dramatically in most cases when any two parameters are changed at the same time. It is also testified that CPGANN has good robustness that adapts well to changes in parameters and can make adjustment to output more stable classification results.

E. ABLATION STUDY

Two more ablation experiments were conducted to compare the performance in this section. The aim of this part is to

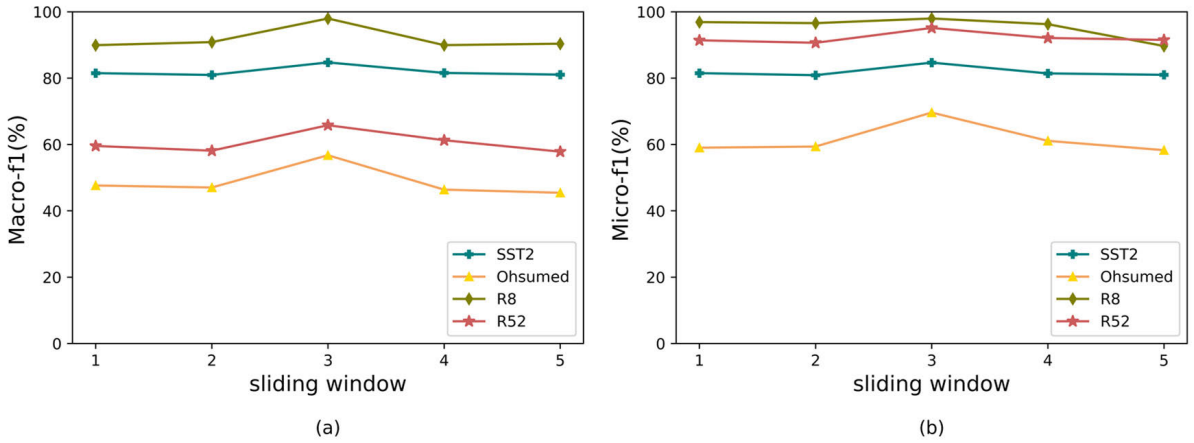


FIGURE 6. Performance of different sliding window sizes on the CPGANN. (a) Macro-F1 (b) Micro-F1.

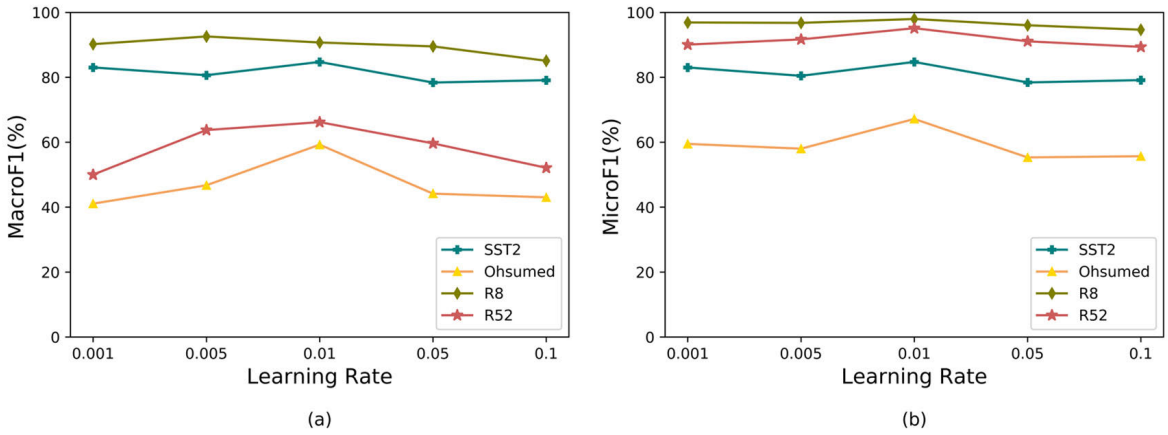


FIGURE 7. Performance of different learning rates on the CPGANN. (a) Macro-F1 (b) Micro-F1.

confirm whether the attention mechanism plays a role in the model and how significant the role is.

1) THE INFLUENCE OF ATTENTION MECHANISM IN AGFE

We propose a variant CPGANN-NA to verify the influence of attention mechanism on feature extraction in the gating unit. NA denotes that there is no attention mechanism in the gating unit and only contains a reset gate and a update gate for information propagation. The comparison results are displayed in V in comparison with the original model CPGANN. We discover that the results are not as good as those of CPGANN with the attention mechanism, which may be due to the fact that the original gating mechanism controls only the selection of individual information and the elements are independent, while the attention mechanism selects the most valuable elements after comparing multiple contextual information. It can be seen that the attention mechanism in the gating unit can improve the performance in the feature extraction and promote the effective aggregation of neighbor information.

2) THE INFLUENCE OF ATTENTION MECHANISM IN EXTRACTING KEYWORDS

We proposed a variant CPGANN-TF to attest the influence of attention mechanism on the effect of extracting keywords, which calculate the term frequency of words without using the attention mechanism. The words are then ranked from highest to lowest frequency, and those with high scores are selected as keywords. The comparison results are presented in 0. It is obvious that the results are worse than the method of using attention to extract keywords. The words with high term frequency that CPGANN-TF selects do not necessarily mean that they are most relevant to the corresponding document, while the attention mechanism assigns weights based on the degree of contribution of words to the central node. Therefore, the extracted keywords are more differentiated for text classification.

From Fig.9, we can see more distinctly that the performance of the two variants is worse than that of CPGANN. This evidences that the attention mechanism does produce a marked effect in calculating node feature representation

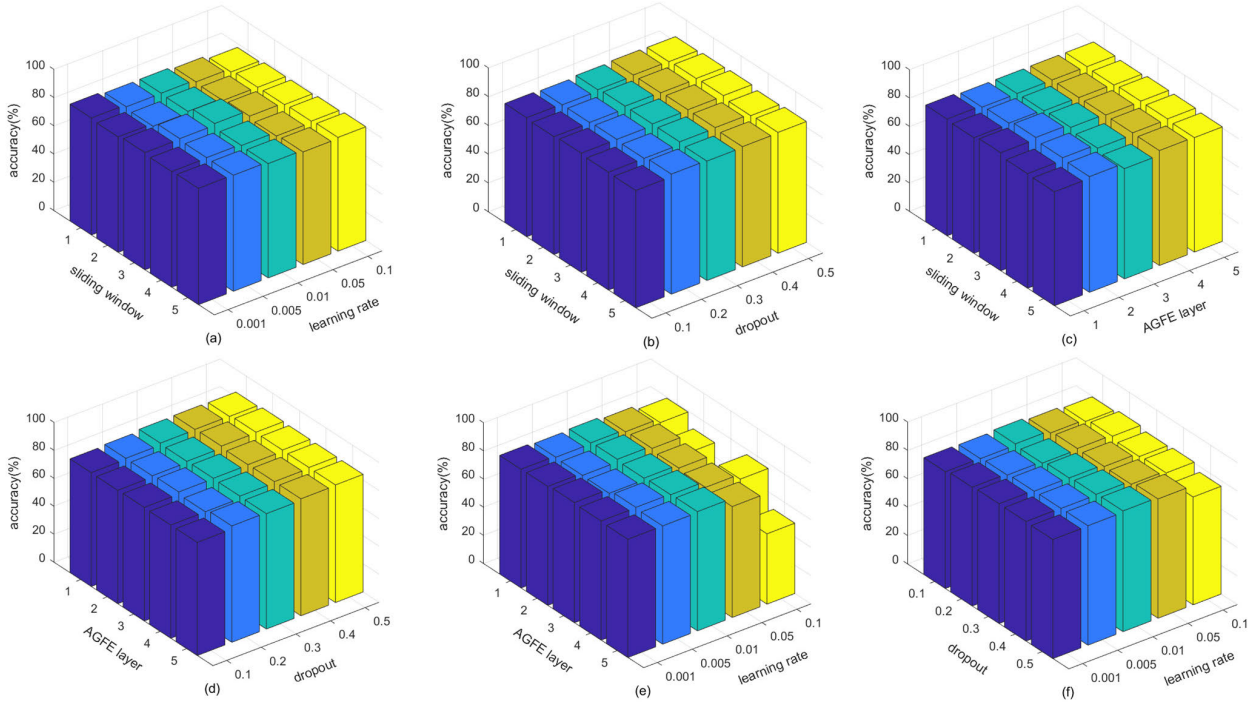


FIGURE 8. The impact of changing any two parameters simultaneously on SST2 dataset.

TABLE 3. Test accuracy (%) comparisons of our method with the variant CPGANN-NA that without attention mechanism in the gating unit on four datasets. The results were reported with mean ± standard deviation.

Model	SST2	R8	R52	Ohsumed
CPGANN-NA	78.86±0.69	96.07±0.11	89.56±0.53	62.50±1.41
CPGANN	**84.67±0.41**	**97.97±0.28**	**95.12±0.69**	**69.61±0.21**

TABLE 4. Test accuracy (%) comparisons of our method with the variant CPGANN-TF that adopting term frequency of word to select keywords on four datasets. The results were reported with mean ± standard deviation.

Model	SST2	R8	R52	Ohsumed
CPGANN-TF	80.88±1.21	83.82±1.61	87.44±0.78	61.06±0.16
CPGANN	**84.67±0.41**	**97.97±0.28**	**95.12±0.69**	**69.61±0.21**

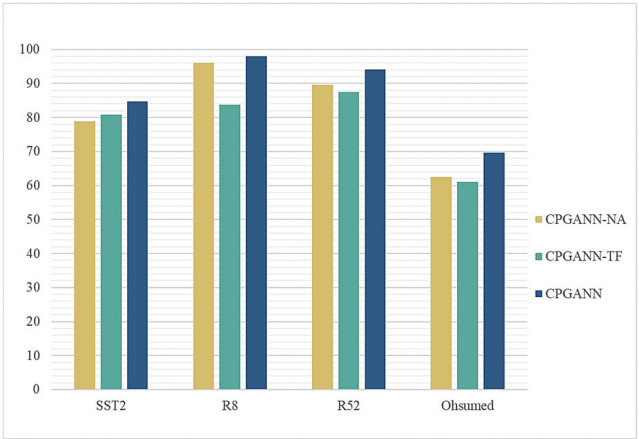


FIGURE 9. Comparison of accuracy (%) between CPGANN and two variants on four datasets.

and extracting keywords, enabling effective integration of node contextual information. It can also more effectively interact with neighbor nodes, and generate more accurate feature embeddings. Besides, the attention mechanism also enhances the expression and propagation of semantics, and the extracted keyword nodes are more discriminative for classification tasks. In conclusion, the adoption of attention mechanism can indeed improve the classification efficiency of the model to some extent.

V. CONCLUSION

This paper proposes a novel gated graph neural network with attention mechanism for text classification in the framework of coupled P systems. The improved network can not only employ the characteristics of independent operation between cells of cell-like P systems but also make use of the mechanism of information exchange and transmission through synapses of tissue-like P systems. The parallelism of the coupled P systems is exploited in general to improve model performance. Specifically, we process the text data into graph structure and put it into an attention-based gated neural network for feature extraction. The reset gate is designed for

forgetting irrelevant information, the update gate is designed for remembering useful information, and an attention mechanism is utilized for assigning different weights to neighbor feature information to generate more accurate node features. Since the features may contain some redundant and irrelevant information at this time, we design to extract keyword nodes by applying the attention mechanism before readout. Experimental results on all datasets prove that CPGANN is superior to other state-of-the-art algorithms. It also testifies that the attention mechanism can produce a notable influence in integrating contextual information and enhance semantic expression to a large extent. Furthermore, we need to consider the time series problem of the text in future studies and take the position order of the text into account for classification.

REFERENCES

[1] X. Yu, X. Ye, and S. Zhang, "Floating pollutant image target extraction algorithm based on immune extremum region," *Digit. Signal Process.*, vol. 123, Apr. 2022, Art. no. 103442, doi: 10.1016/j.dsp.2022.103442.

[2] X. Yu and X. Tian, "A fault detection algorithm for pipeline insulation layer based on immune neural network," *Int. J. Pressure Vessels Piping*, vol. 196, Apr. 2022, Art. no. 104611, doi: 10.1016/j.ijpvp.2022.104611.

[3] X. Yu, Y. Lu, and Q. Gao, "Pipeline image diagnosis algorithm based on neural immune ensemble learning," *Int. J. Pressure Vessels Piping*, vol. 189, Feb. 2021, Art. no. 104249, doi: 10.1016/j.ijpvp.2020.104249.

[4] K. He, X. Cao, Y. Shi, D. Nie, Y. Gao, and D. Shen, "Pelvic organ segmentation using distinctive curve guided fully convolutional networks," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 585–595, Feb. 2019, doi: 10.1109/TMI.2018.2867837.

[5] W. Y. Ying, L. Zhang, and H. L. Deng, "Sichuan dialect speech recognition with deep LSTM network," *Frontiers Comput. Sci.*, vol. 14, no. 2, pp. 378–387, Apr. 2020, doi: 10.1007/s11704-018-8030-z.

[6] P. Li, H. Wang, K. Q. Zhu, Z. Wang, X. Hu, and X. Wu, "A large probabilistic semantic network based approach to compute term similarity," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2604–2617, Oct. 2015, doi: 10.1109/TKDE.2015.2419673.

[7] W. Hua, Z. Wang, H. Wang, K. Zheng, and X. Zhou, "Understand short texts by harvesting and analyzing semantic knowledge," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 3, pp. 499–512, Mar. 2017, doi: 10.1109/TKDE.2016.2571687.

[8] Z. Chen, T. Cai, C. Chen, Z. Zheng, and G. Ling, "SINE: Side information network embedding," Presented at the Int. Conf. Database Syst. Adv. Appl., Apr. 2019, doi: 10.1007/978-3-030-18576-3_41.

[9] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1–12, doi: 10.1145/3219819.3219947.

[10] Z. Liu, V. W. Zheng, Z. Zhao, F. Zhu, K. C.-C. Chang, M. Wu, and J. Ying, "Semantic proximity search on heterogeneous graph by proximity embedding," Presented at the AAAI, San Francisco, CA, USA, Feb. 2017.

[11] H. Chen, H. Yin, X. Sun, T. Chen, B. Gabrys, and K. Musial, "Multi-level graph convolutional networks for cross-platform anchor link prediction," Presented at the KDD, Aug. 2020, doi: 10.1145/3394486.3403201.

[12] H. Cai, S. Lv, G. Lu, and T. Li, "Graph convolutional networks for fast text classification," in *Proc. 4th Int. Conf. Natural Lang. Process. (ICNLP)*, Mar. 2022, pp. 420–425, doi: 10.1109/ICNLP55136.2022.00078.

[13] J. Cheng, Q. Wang, Z. Tao, D. Xie, and Q. Gao, "Multi-view attribute graph convolution networks for clustering," Presented at the 29th Int. Joint Conf. Artif. Intell. 17th Pacific Rim Int. Conf. Artif. Intell., Jul. 2020.

[14] X. W. Liu, X. Zhu, M. Li, L. Wang, E. Zhu, T. Liu, M. Kloft, D. Shen, J. Yin, and W. Gao, "Multiple kernel k-means with incomplete kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1191–1204, May 2020, doi: 10.1109/TPAMI.2019.2892416.

[15] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009, doi: 10.1109/TNN.2008.2005605.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," Presented at the 5th Int. Conf. Learn. Represent., 2016.

[17] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[18] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," 2015, *arXiv:1511.05493*.

[19] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," Presented at the Int. Conf. Learn. Represent., 2018.

[20] Z. Y. Wu, D. C. Pi, J. F. Chen, M. Xie, and J. J. Cao, "Rumor detection based on propagation graph neural network with attention mechanism," *Expert Syst. Appl.*, vol. 158, pp. 1–16, Nov. 2020, doi: 10.1016/j.eswa.2020.113595.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[22] F. Enríquez, J. A. Troyano, and T. López-Solaz, "An approach to the use of word embeddings in an opinion classification task," *Expert Syst. Appl.*, vol. 66, pp. 1–6, Dec. 2016, doi: 10.1016/j.eswa.2016.09.005.

[23] S. Hirschmeier, J. W. Melsbach, D. Schoder, and S. Stahlmann, "Improving recall and precision in unsupervised multi-label document classification tasks by combining word embeddings with TF-IDF," in *Proc. Eur. Conf. Inf. Syst.*, 2020, pp. 1–10. [Online]. Available: https://aisel.aisnet.org/ecis2020_rp/99

[24] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751, doi: 10.3115/v1/D14-1181.

[25] R. Johnson and T. Zhang, "Supervised and semi-supervised text categorization using LSTM for region embeddings," Presented at the 33rd Int. Conf. Mach. Learn., New York, NY, USA, 2016.

[26] G. Păun, "Computing with membranes," *J. Comput. Syst. Sci.*, vol. 61, no. 1, pp. 108–143, Aug. 2000, doi: 10.1006/jcss.1999.1693.

[27] B. Song, X. Luo, L. Valencia-Cabrera, and X. Zeng, "The computational power of cell-like P systems with one protein on membrane," *J. Membrane Comput.*, vol. 2, no. 4, pp. 332–340, Dec. 2020, doi: 10.1007/s41965-020-00063-x.

[28] H. Peng, J. Wang, P. Shi, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "Fault diagnosis of power systems using fuzzy tissue-like P systems," *Integr. Comput. Aided Eng.*, vol. 24, no. 3, pp. 1–11, 2017, doi: 10.3233/ICA-170552.

[29] H. Peng, B. Li, J. Wang, X. Song, T. Wang, L. Valencia-Cabrera, I. Pérez-Hurtado, A. Riscos-Néñez, and M. J. Pérez-Jiménez, "Spiking neural P systems with inhibitory rules," *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 105064, doi: 10.1016/j.knosys.2019.105064.

[30] T. Wu, L. Pan, and A. Alhazov, "Computation power of asynchronous spiking neural P systems with polarizations," *Theor. Comput. Sci.*, vol. 777, pp. 474–489, Jul. 2019, doi: 10.1016/j.tcs.2018.10.024.

[31] X. Liu, Y. Zhao, and W. Sun, "Tissue P systems with cooperating rules," *Chin. J. Electron.*, vol. 27, no. 2, pp. 324–333, Mar. 2018, doi: 10.1049/cje.2018.01.005.

[32] T. Wu, Y. Wang, S. Jiang, Y. Su, and X. Shi, "Spiking neural P systems with rules on synapses and anti-spikes," *Theor. Comput. Sci.*, vol. 724, pp. 13–27, May 2018, doi: 10.1016/j.tcs.2017.12.015.

[33] F. Bernardini, M. Gheorghe, N. Krasnogor, and R. C. Muniyandi, "On P systems as a modelling tool for biological systems," in *Proc. Int. Workshop Membrane Comput.*, vol. 3850, 2005, pp. 114–133, doi: 10.1007/11603047_8.

[34] W. Liu, "A fault diagnosis method for power transmission networks based on spiking neural P systems with self-updating rules considering biological apoptosis mechanism," *Complexity*, vol. 2020, pp. 1–12, Jan. 2020, doi: 10.1155/2020/2462647.

[35] R. Blanco and C. Lioma, "Graph-based term weighting for information retrieval," *Inf. Retr.*, vol. 15, no. 1, pp. 54–92, Feb. 2012, doi: 10.1007/s10791-011-9172-x.

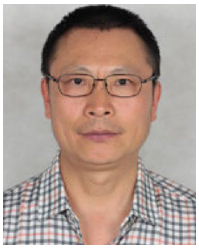
[36] F. Rousseau, E. Kiagias, and M. Vazirgiannis, "Text categorization as a graph classification problem," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, Jan. 2015, pp. 1702–1712, doi: 10.3115/v1/P15-1164.

[37] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," Presented at the 25th Int. Joint Conf. Artif. Intell., New York, NY, USA, Jul. 2016.

[38] D. Shen, G. Wang, W. Wang, M. R. Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin, "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 440–450, doi: 10.18653/v1/P18-1041.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.
[40] L. Huang, D. Ma, S. Li, X. Zhang, and H. Wang, "Text level graph neural network for text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3444–3450, doi: 10.18653/v1/D19-1345.
[41] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 1–13, doi: 10.18653/v1/E17-2068.
[42] S. Qian, J. Wang, J. Hu, Q. Fang, and C. Xu, "Hierarchical multi-modal contextual attention network for fake news detection," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2021, pp. 1–16, doi: 10.1145/3404835.3462871.
[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[44] R. K. Yadav, L. Jiao, O.-C. Granmo, and M. Goodwin, "Enhancing interpretable clauses semantically using pretrained word representation," in *Proc. 4th BlackboxNLP Workshop Analyzing Interpreting Neural Netw. NLP*, 2021, p. 10, doi: 10.18653/v1/2021.blackboxnlp-1.19.
[45] D. Q. Nguyen, T. D. Nguyen, and D. Phung, "Universal graph transformer self-attention networks," in *Proc. Companion Web Conf.*, Apr. 2022, pp. 1–20, doi: 10.1145/3487553.3524258.
[46] Y. Gu, Y. Wang, H. Zhang, J. Wu, and X. Gu, "Enhancing text classification by graph neural networks with multi-granular topic-aware graph," *IEEE Access*, vol. 11, pp. 20169–20183, 2023, doi: 10.1109/ACCESS.2023.3250109.
[47] Z. Deng, C. Sun, G. Zhong, and Y. Mao, "Text classification with attention gated graph neural network," *Cogn. Comput.*, vol. 14, pp. 1464–1473, Apr. 2022.
[48] Q. Qin, W. Hu, and B. Liu, "Feature projection for improved text classification," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1–6, doi: 10.18653/v1/2020.acl-main.726.
[49] Y. Zhang, X. Yu, Z. Cui, S. Wu, Z. Wen, and L. Wang, "Every document owns its structure: Inductive text classification via graph neural networks," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 334–339, doi: 10.18653/v1/2020.acl-main.31.

JIAQI ZHANG received the bachelor's degree from Shandong Normal University, China, in 2021. She is currently pursuing the master's degree with the Business Department, Shandong Normal University. Her research interests include graph neural networks, membrane computing, and data mining.



XIYU LIU (Member, IEEE) received the Ph.D. degree in mathematical sciences from Shandong University, in 1990. He was a Visiting Researcher with The Hong Kong Polytechnic University, from 2001 to 2003. He is currently a Professor, a Ph.D. Supervisor, and the Dean of the Institute of Management Science, Shandong Normal University. He has published three works, more than 230 SCI and EI retrieval papers, presided over four National Natural Science Foundation projects, completed a number of international cooperation projects of Hong Kong Polytechnic University, major projects of the Natural Science Foundation of Shandong Province, and major soft science projects of Shandong province. He is one of the first Taishan Distinguished Professors, the Discipline Leader of management science and engineering, a Young and Middle-Aged Expert with outstanding contributions in Shandong Province, and the Executive Director of the National Society of Management Science and Engineering. His research interests include membrane computing, computational intelligence, data mining, and knowledge engineering.