GCN-BERT and Memory Network Based Multi-Label Classification for Event Text of the Chinese Government Hotline

BIN LIU
Anhui Business College, Wuhu 241000, China
e-mail: liubin@abc.edu.cn

This work was supported in part by the Anhui Province Quality Engineering Project under Grant 2020jxtd093, and in part by the Key Research Project of Natural Science in Anhui Province under Grant KJ2020A1082.

ABSTRACT In order to automatically generate multiple labels for the event text of the Chinese government hotline, this paper proposes a multi-label classification framework based on graph convolutional network (GCN), BERT, and memory network. The framework consists of three modules: label count prediction module, label semantic insert module, and label selection module. In the label count prediction module, this paper constructs the event graph with the abstract meaning representation (AMR) and extracts the event topic information vector with GCN. To predict the label count, this paper first uses BERT to extract the event semantic information vector and then fuses it with the event topic information vector (GCN-BERT fusion vector) with a dynamic fusion gate. In the label semantic insert module, the event label candidate set is obtained using a multi-hop memory network to store the event label semantic information, and then uses the answer selection framework, which matches the GCN-BERT fusion vector with the event label semantic memory vector. In the label selection module, this paper uses the label count based multi-label selection to sort the event label candidate set and guide to output the optimal multi-label set of the event. Comparison experimental results show that the proposed framework outperforms all baselines and ablation studies demonstrate the effectiveness of each module.

INDEX TERMS The Chinese government hotline, multi-label classification, GCN, BERT, memory network.

I. INTRODUCTION The Chinese government hotline records public demands in the form of event text. For a hotline operator, he analyzes the event text and labels the event. Traditional text classification methods mainly focus on a single label and fit poorly for the event text of the Chinese government hotline as a hotline event may have multiple labels [1]. Furthermore, the event text presents continuous expansion characteristics that cover all aspects of urban life and realize intelligent event classification. Hence, it is difficult for an unprofessional hotline operator to label the events. Generally, the length of different event text varies greatly and the content is sparse [2]. Word2vec, GloVe, and other word vector models can collect simple contextual information from the event text, but they cannot pay attention to the context-aware information and have trouble dealing with the polysemy of one word in natural languages [3], [4]. Pre-trained language models (BERT, XLNet, RoBERTa, etc. [5], [6], [7]) containing a large amount of priori knowledge can effectively solve this problem. However, due to truncating the input text, which may cause semantic information loss, pre-trained language models are insufficient to process long texts well. This problem can be solved by constructing the event graph with an AMR model, which can effectively construct the topic information of the event text, and then applying the GCN to nodes and edges of constructed graph [8], [9]. At the same time, the memory network can store the semantic information of classification labels effectively, thus improving the accuracy of ‘event-labels’ matching [10]. The event text label count can provide
a priori knowledge for event multi-label classification to determine the size of the final labels set. This paper proposes a multi-label classification model for the Chinese government hotline. Our contributions are as follows:

(1) This paper proposed a novel framework for multi-label classification, which labels event text by answer selection and predicts event label count with prior knowledge about labels’ text allowing for a more accurate generation of event labels.

(2) To solve the problems of different event text lengths and unclear elements, this paper constructs an AMR-based graph and use GCN to obtain topic information of the event text, which can effectively solve the problem of long event text. To further address the issue, a gate mechanism is used to dynamically fuse event text information extracted by BERT and event topic information extracted by GCN.

(3) This paper proposes an event label generation module based on semantic information and a multi-hop memory network, which can effectively match the event representation vector and label semantic vector by storing the event label semantic information in the memory network.

The rest of the paper is organized as follows. Section 2 contains the literature review of several related tasks including multi-label classification, graph convolution network, and memory network. Section 3 contains the overall structure of the proposed model and its details. Section 4 contains the experiments and results with analysis. Section 5 contains the conclusions and discussions.

II. RELATED WORK

A. MULTI-LABEL CLASSIFICATION

Multi-label classification task predicts labels of multiple categories, and the labels are not mutually exclusive, so the label space is complicated. Yang proposed that a multi-label classification task is regarded as a sequence generation problem, and a sequence generation model with a novel decoder structure is adopted to solve this problem. This method can capture the correlations between labels and select the most informative words automatically when predicting different labels. However, the prediction accuracy is not stable when the number of real labels is uncertain [11]. Lin proposed a multi-label text classification model based on sequence-to-sequence learning, which generates higher-level semantic unit representation through multi-level dilated convolution and a hybrid attention mechanism. This method can extract both the information at the word level and the level of the semantic unit. However, this method does not consider the association between sentences or labels [12]. Lanchantin proposed the Label Message Passing (LaMP) Neural Networks, which models the joint prediction of multiple labels. The neural network has an excellent performance in the prediction of intensive labels, but this method still does not effectively leverage the semantic correlation between sentences [13]. Hong proposed to use a neural tensor network to explore the correlations between the labels of neighbors and classify query instances with these correlations. The correlations and interdependence between labels are made full use of while other studies research from the perspective of label relevance [14]. Wang et al. proposed an adversarial learning framework to strengthen the similarity between the joint distribution of the ground truth multi-labels and the predicted multiple labels [15]. Meanwhile, Xu et al. proposed a hierarchical method to utilize the label correlation of different branches to facilitate the differentiation of hierarchical multi-label classification [16]. For multi-relationship message delivery, Ozmen et al. realized multiple types of bi-directional relationships modeling between labels, which improved the mining of valuable information in labels [17].

B. GRAPH CONVOLUTION NETWORK

Graph convolutional neural network extracts features from an abstract graph and generates corresponding representation vectors, which is widely used in many fields related to graph including text classification. Yao proposed to use graph convolutional networks for text classification. They built a single text graph for a corpus based on word co-occurrence and document word relations, then constructed a text graph convolutional network (Text GCN) for the corpus. However, the model prediction accuracy is often affected by the corresponding corpus size [18]. Lei et al. proposed to learn and fuse multi-hop neighbor information relationships, which adopts the weight-sharing mechanism to design different order graph convolutions for avoiding the potential concerns of over-fitting [19]. Tang et al. proposed an improved graph convolutional network (IGCN), which uses the short-range contextual dependency and the long-range contextual dependency captured by the dependency relationship together to address the problem of contextual dependency [20]. It is worth noticing that these methods can effectively address contextual dependency and lexical polysemy, but they tend to perform better on long texts than on short ones. To improve the performance of GCN-based text classification models for different length texts, several improved studies have been proposed. Huang et al. proposed to build graphs for each input text with global parameters sharing instead of a single graph for the whole corpus [21]. Xu proposed an improved sequence-based feature propagation scheme, which overcomes the limitations of textual features in short texts by fully uses word representations and word interactions at the document level [22]. For further utilize the potential correlation information between words, Deng et al. proposed a novel graph-based model where every document is represented as a text graph, which can capture semantic correlations between the words [23].

C. MEMORY NETWORK

Traditional deep learning models (such as RNN, LSTM, and GRU) take hidden states or attention mechanism as the memory module, but their short-term memory is impossible to accurately record the full content of a paragraph. To address the problem, Weston proposed a memory network as reason with inference components combined with a long-term memory component. The long-term memory can be read and
written, to use it for prediction. But the model was not easy to train via backpropagation and required supervision at each layer of the network [24]. So, in subsequent studies, a series of improved models based on the memory network framework is proposed and applied to a variety of scenarios. By combining with the end-to-end model, the flexibility of the model allows us to apply it to tasks as diverse as (synthetic) question answering and language modeling [24], [25], [26], [27], [28]. At the same time, hierarchical memory networks proposed a new memory data structure based on clustering to accelerate the training speed, and a new search method to adapt to this structure for scoring calculation [29]. Although the above works realize the end-to-end training of the memory network, some details will be lost when the model only encodes the input at the sentence level. To solve it, Xu et Al. proposed a novel hierarchical memory network, which realized the modeling of word-level and sentence-level to get the integrated output of sentences and words [30].

III. MODEL DESIGN
A. PROBLEM STATEMENT
Firstly, this paper state the multi-label event classification problem definition. Given a series of event texts \( D = \{d_1, d_2, \ldots, d_m\} \), where every event text contains a variable size of \( n \) sentences \( d_i = \langle w_1, w_2, \ldots, w_n \rangle \). Define a classification label list \( C = \langle c_1, c_2, \ldots, c_k \rangle \) of fixed size \( k \), the ultimate goal of the event classification task is to generate an event label set \( \{l_1, \ldots, l_l\} \), where \( l \in C \) represents the label(s) for the Chinese government hotline event text.

B. MODEL STRUCTURE
1) EVENT LABEL COUNT PREDICTION
The proposed label count prediction model includes 3 modules: (i) AMR-based Graph Construction; (ii) Feature extraction based on GCN and BERT; (iii) Label count prediction layer. The overall construction of the proposed model is shown in Figure 1.

a: AMR BASED GRAPH CONSTRUCTION
In this section, this paper introduces how to use AMR to construct the event key information interaction graph from an event text. AMR established a unified annotation standard, which can solve the argument-sharing problem due to a noun dominated by multiple predicates and effectively express the theme of the sentence. Therefore, this paper use AMR to build the key information interaction diagram of the event. Since the AMR parsing algorithm is not the main point of this paper, so no details are introduced about this algorithm process. This paper use a seq2seq-based model in literature to construct the AMR graph of event texts, and the generated AMR graph is embedded as the input of GCN [31].

Given an event text, this paper first uses the sentence segmentation method to split the event text into sentences list of size, then do word segmentation on the sentences list through off-the-shelf tools such as Stanford CoreNLP to get the overall words that appeared in the event text corpus. Then, the word segmentation results are sent into the seq2seq-based AMR model to construct the AMR graph corresponding to each sentence. Using AMR graph directly for graph embedding is tedious for this task. On the one hand, AMR has many different predicates, which correspond to different edge labels. While explicitly grouping edge labels into a single one can effectively reduce the difficulty in the graph embedding stage, it is not an ideal solution since it incurs in loss of information [32], [33]. On the other hand, the latent information in edges can depend on the context in which they appear in the graph, which makes it very difficult to obtain the hidden semantic information from these labeled edges.
edges. To solve this problem, this paper converts the AMR graph into an unlabeled bipartite graph by using Levi Transformation, which turns original edges into nodes creating an unlabeled graph [34]. The new graph can directly represent the relationships between concept nodes and relations in the original AMR graph, which is very useful for embedding edges from the new graph and extracting hidden information of relations. After graph embedding, this paper can employ different GNNs. AMR graph transformation is described as algorithm 1, which is an improved algorithm based on Levi Transformation. An example of algorithm 1 is shown in Figure 2.

**Algorithm 1 AMR Graph Transformation**

**Input:** A rooted and directed AMR graph $G = (V, E, R)$, $v \in V$ indicates concept nodes, $r \in R$ indicates relation type

**Output:** An unlabeled and undirected graph $G_i = (V_i, E_i)$

**Step 1:** Construct triples $(v_i, r, v_j)$ from the graph G in order of breadth priority, $(v_i, r, v_j)$ can indicate labeled edges.

**Step 2:** Disassemble triples $(v_i, r, v_j)$ into tuples by using Levi Transformation and transfer relations $r \in R$ into nodes of new graph $(v_i, r, v_j) \rightarrow (v_i, r), (r, v_j)$

**Step 3:** Connect tuples group to form a directed graph according to the hierarchical order of the original graph

$$G_i'(V_i, E_i) = \text{Concat}(\text{Concat}(v_i, r), \text{Concat}(r, v_i))$$

**Step 4:** Convert directed edges in the graph $G_i'$ into undirected edges to form the output graph $G_i = (V_i, E_i)$

After Levi Transformation, all obtained n AMR graphs are transformed to n undirected graph. When $n = 1$, the event key information interaction graph is the undirected graph of AMR transformation. When $n > 1$, what the model get now is n unrelated graphs that only have relations and nodes in the sentence, but lack relations and nodes between sentences. So, the next step is to build relations between sentences. The proposed model now adopt a simple strategy, that is, if two nodes $v^m_i$ and $v^n_j$ in different AMR graphs of the same event appear in the same sentence, it implies that they are related, so this paper add an edge between $v^m_i$ and $v^n_j$. Algorithm 2 shows the construction process. The relation between sentences is obtained through algorithm 2, and then the interaction graph of event key information is completed by combining the relation in sentences.

**Algorithm 2 Construction of Event Graph**

**Input:** The event text $D$

**Output:** Event graph

1. Do sentence segmentation on $D$ to get the sentences list $S = \{s_1, s_2, \ldots, s_k\}$
2. Do word segmentation on every sentence to get word lists $s^\text{seg}_i = \{w^1_i, w^2_i, \ldots, w^l_i\}$, where $i = 1, 2, \ldots, k$
3. If $n$ equals to 1 then
4. Do Seq2Seq-AMR on $s_1$ to get the AMR graph $G$ and vertex list $s'_1 = [v^1_1, v^1_2, \ldots]$
5. Else
6. For sentence $s$ in $S$ do
7. Do Seq2Seq-AMR on $s_i$ to get the AMR graph $G_i$ and vertex list $s'_i = [v^1_i, v^2_i, \ldots]$
8. End for
9. For vertex $v^m_i$ and $v^n_j$ do
10. If $v^m_i$ and $v^n_j$ in the same sentence then
11. Add an edge between $v^m_i$ and $v^n_j$
12. Else
13. Continue
14. End if
15. End for
16. End if

**b: FEATURE EXTRACTION BASED ON GCN AND BERT**

**i) VERTEX ENCODER**

For the event text $D$ processed by word segmentation, this paper use three different vector representation methods to process it. First, each token in the event text is processed by word embedding, and then get the word embedding sequence of the text:

$$W = \{w^1, w^2, \ldots, w^k, \ldots, w^l, \ldots\}$$  \hspace{1cm} (1)

where $w^k$ is the vertex token word embedding, $w^l$ is the non-vertex token word embedding, $w^k$ and $w^l$ share the same embedding table. Second, in order to obtain the sentence-level information of event text $D$, this paper use segment embedding to process it and then get the sentence features embedding sequence:

$$S = \{s_1, s_2, \ldots, s^k_1, \ldots, s_j, \ldots\}$$  \hspace{1cm} (2)
where $s_i$ is the vertex token segment embedding, $s_j$ is the non-vertex token segment embedding. Thirdly, to obtain the position information of each token, the event text is processed by position embedding. The position embedding sequence of the text is as follows:

$$P = \{p_1, p_2, \ldots, p_i, \ldots, p_j, \ldots\} \quad (3)$$

where $p_i$ and $p_j$ are the vertex token and non-vertex token position embedding respectively. Finally, the final non-vertex embedding and vertex embedding are the sum of their corresponding word embedding, segment embedding and position embedding respectively:

$$e_i = w_j + s_i + p_i \quad (4)$$

$$e_i^k = w^k_j + s^k_i + p^k_i \quad (5)$$

where $e_i$ and $e_i^k$ are the final embedding vectors of non-vertex and vertex respectively. So, the final embedding result is as follows:

$$E = \{e_1, e_2, \ldots, e_i, \ldots, e_j, \ldots\} \quad (6)$$

The vertex representation vector $v_i$ is obtained by the final embedding set $E$ through a self-attention mechanism:

$$v_i = \alpha_i^k e_i^k \quad (7)$$

$$\alpha_i^k = \frac{\exp(u^k_i u_w)}{\sum_i \exp(u^k_i u_w) + \sum_j \exp(u_j u_w)} \quad (8)$$

$$u_i^k = (\tanh(W_k e_i^k + b_k))^T \quad (9)$$

$$u_j = (\tanh(W_w e_j + b_w))^T \quad (10)$$

where $W_k, b_k, W_w, b_w$ are the learnable parameter, $u_w$ is the context vector.

### ii) GRAPH CONVOLUTION

After the input graph is processed with vertex encoder, a vertex representation vector set is obtained:

$$V = \{v_1, v_2, \ldots, v_i, \ldots\} \quad (11)$$

where $v_i$ is the vertex representation vector of $i$-th vertex. For the vertex vector set $V$, this paper use GCN network to process it to make use of the graph structure of the constructed graph. In this paper, the layer-wise propagation rule of Kipf and Welling was used to conduct the GCN model [9]. The event text representation vector $V_{\text{gen}}$ is calculate as follows:

$$V_{\text{gen}} = \hat{\Delta} \text{ReLU}(\hat{\Delta}W^{0})W^{1} \quad (12)$$

$$\hat{\Delta} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} \quad (13)$$

where $\hat{A} = A + I_N$ is the adjacency matrix of the constructed graph with added self-connections, $I_N$ is the identity matrix. $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ is the degree matrix and $W^{0}$, $W^{1}$ are trainable input-to-hidden weight matrix and hidden-to-output weight matrix respectively.

### iii) TEXT REPRESENTATION BASED ON BERT

After the input event text is constructed into a graph structure by AMR and then feed it to GCN to get the representation vector, which usually contains the semantic structure information and topic information of the event text. But it is difficult to effectively obtain more accurate semantic information such as character, word and context information. Therefore, this paper use BERT to process the event text to further obtain the character, word and context information of the event text, so as to improve the accuracy and robustness of the model. For the given event text $D$, the text representation vector $V_{\text{lm}}$ is calculate as follows:

$$V_{\text{lm}} = f(D) \quad (14)$$

where $f$ is the BERT language model.

### c: LABEL COUNT PREDICTION LAYER

To take full advantage of the GCN and BERT language models, this paper fuse GCN representation vector $V_{\text{gen}}$ and language model representation vector $V_{\text{lm}}$ with a gating mechanism to obtain a fused contextual semantic information representation $V_{\text{fuse}}$. The calculation formula is as follows:

$$V_{\text{fuse}} = (\lambda_1 \odot V_{\text{gen}}) + (\lambda_2 \odot V_{\text{lm}}) \quad (15)$$

where $\lambda_1$, $\lambda_2$ are the fusion weights of GCN information and language model information respectively.

A multi-layer perceptron (MLP) network is used for the label count prediction. This network consists of a fully-connected layer and an output layer with the softmax function for classification, the probability distribution of event label count is calculated as follows:

$$p_{\text{count}} = \text{softmax}(W_{\text{mlp}}V_{\text{lm}} + b_{\text{mlp}}) \quad (16)$$

where $W_{\text{mlp}}$ is the learnable weight matrix, $b_{\text{mlp}}$ is the bias vector, and $p_{\text{count}}$ is the event category probability.

### 2) EVENT LABEL SEMANTIC INSERT

The label semantic insert module in this paper is mainly composed of BERT-based hierarchical memory network and attention mechanism. The module matches the event representation vector with the label semantic vector to obtain the label candidate set. BERT captures a rich hierarchy of linguistic information, with surface features in lower layers, syntactic features in middle layers and semantic features in higher layers [35]. Therefore, this paper proposes a BERT-based 3-hop memory network, which leverages different layers in BERT to extract surface information, syntactic information and semantic information of ‘sanding’, and then different level information is combined with different memory network to storage more text information.

As shown in Figure 1, Given label text set, BERT model is used to process all elements in set $X$. Two low layers (layer 1, 2, 3 and 4) output vector sets $A_l = \{a_i | i = 1, \ldots, n\}$ and $C_l = \{c_i | i = 1, \ldots, n\}$ of BERT are taken as surface input
In the event label count prediction module, the multi-classification cross-entropy loss function is used during the model training:

$$L_{\text{count}} = - \sum_{j} y_{j}^\text{count} \log p_{j}^\text{count}$$  \hspace{1cm} (23)$$

where $p_{j}^\text{count}$ is the predicted classification probability of event $j$, and $y_{j}^\text{count}$ is the indicator variable (0 or 1).

In the event label semantic insert module, the output vectors $u_t$, $u_m$, $u_h$ of Attention are processed to get the label candidate set:

$$p_{as} = \text{softmax}(W_{as}[u_t, u_m, u_h] + b_{as})$$  \hspace{1cm} (24)$$

where $W_{as}$ is learnable matrices, $b_{as}$ is bias vectors and $p_{as}$ is event matching probabilities.

The standard cross-entropy loss function is used for the event label semantic insert model:

$$L_{as} = - \sum_{i} [y_i \log p_i + (1 - y_i) \log(1 - p_i)]$$  \hspace{1cm} (25)$$

where $p_i$ is the event-label matching probability, and $y_i$ is the indicator variable (0 or 1).

In this paper, the joint loss function strategy is used to optimize the whole network, and the final objective is to minimize the above two loss functions:

$$L = \lambda_1 L_{\text{count}} + \lambda_2 L_{as}$$  \hspace{1cm} (26)$$

where $\lambda_1$ and $\lambda_2$ are the hyper-parameters to balance two losses.

4) EVENT LABEL SELECTION

In the prediction stage, for the input event text $D$, this paper use a label selection method to improve the performance of multi-label classification model, the specific steps are as follows: Firstly, the Label Count Prediction module is used to predict the label count $k$ of the event text. Secondly, the matching probability values between the current event representation vector and all 377 label semantics are calculated with the model, and then get a matching probability list $L = \{p_1, p_2, \ldots, p_{377}\}$. Finally, the match probability list $L$ is sorted by values, and the label set corresponding to the first $k$ maximum probabilities of list $L$ is taken as the multi-label of input event.

### IV. EXPERIMENT

A. DATASET

1) DATASET CONSTRUCTION

This paper constructed an experimental dataset based on the real event assignment case of Wuhu government hotline [39]. The dataset includes two fields: “Event text” and “Event label”. “Event text” is the overall description of events based on message of the party concerned. “Event label” is the labels involved in the event text to describe different event labels, including 377 different classification labels. For the whole dataset, there are 30000 multi-label event texts, consisting of 2796 single-label events and 27204 multi-label events. The datasets description is shown in Table 1.

### TABLE 1. Detailed information of dataset. (“documents” is the number of events. “sentences” is the total number of sentences in dataset. “characters” is the total number of characters in dataset. “labels” is the total number of labels. “cardinality” is the average number of labels per event. “density” is equal to cardinality/labels. “diversity” is the number of distinct label sets appeared in the data set.)

| documents | sentences | characters | labels | cardinality | density | diversity |
|-----------|-----------|------------|--------|--------------|---------|-----------|
| 30,000    | 102,673   | 4,853k     | 377    | 2.1819       | 0.0058  | 929       |

Like surface level memory and syntactic level memory, two middle layers (layer 5, 6, 7 and 8) output vector sets $A_m = \{a^m_i | i = 1, \ldots, n\}$ and $C_m = \{c^m_i | i = 1, \ldots, n\}$ of BERT is taken as syntactic input memory representation and syntactic output memory representation respectively. The calculation formula of BERT-based syntactic level memory om is as follows:

$$p_i^m = \text{softmax}(u_i^m a_i^m)$$ \hspace{1cm} (19)$$

$$o_m = \sum_i p_i^m c_i^m$$ \hspace{1cm} (20)$$

At the end of label semantic insert module, the corresponding output vectors $u_t$, $u_m$, $u_h$ were obtained from memory vectors $o_t$, $o_m$, $o_h$ through attention mechanism respectively.

3) JOINT LEARNING LOSS FUNCTION

In the event label count prediction module, the multi-classification cross-entropy loss function is used during the model training:

$$L_{\text{count}} = - \sum_{j} y_j^\text{count} \log p_j^\text{count}$$ \hspace{1cm} (23)$$

where $p_j^\text{count}$ is the predicted classification probability of event $j$, and $y_j^\text{count}$ is the indicator variable (0 or 1).

In the event label semantic insert module, the output vectors $u_t$, $u_m$, $u_h$ of Attention are processed to get the label candidate set:

$$p_{as} = \text{softmax}(W_{as}[u_t, u_m, u_h] + b_{as})$$ \hspace{1cm} (24)$$
At the first stage, the proposed model labeled the dataset preliminary with the crowd-sourcing method. Each event was dually annotated by two annotators working independently. At the second stage, the proposed model invited 10 domain experts to give the final labeling results only once. Finally, the proposed model fused results from two stages to get the golden result.

3) DATA QUALITY
This paper randomly selected 500 events as sample and invited three domain experts to relabel these samples independently, and then use these results to evaluate the quality of dataset generated in previous stages. As the benchmark, this paper use Cohen’s Kappa as evaluation metric, which describes the consistency of two results. For event label annotations, Cohen’s Kappa is 71%. The result shows that two annotation results are highly consistent. The Cohen’s Kappa of 61%-80% indicates the substantial agreement, so the annotation result on previous stages are sufficiently reliable for the following experiments.

B. EVALUATION METRICS
To evaluate the performance of the proposed model, five widely-used evaluation metrics are employed.

**Precision:** precision focuses on the proportion of correctly predict labels in predicted set, its calculation formula is as follows

\[
Pr = \frac{1}{N} \sum_{i=1}^{N} \frac{\|\hat{y}_i \cap y_i\|}{\|\hat{y}_i\|} \tag{27}
\]

where \(N\) denotes the number of samples in test set, \(y_i\) denotes the set of ground truth labels associated with sample \(i\), and \(\hat{y}_i\) is the predicted set of labels associated with sample \(i\).

**Recall:** recall is the opposite of precision, which focuses on the proportion of correctly predict labels in ground truth set, its calculation formula is as follows,

\[
Re = \frac{1}{N} \sum_{i=1}^{N} \frac{\|\hat{y}_i \cap y_i\|}{\|y_i\|} \tag{28}
\]

where \(N\), \(y_i\), and \(\hat{y}_i\) are the same as precision calculation formula.

**F1-score:** F1-score can be interpreted as a weighted average of the precision and recall. The calculation formula is as follows,

\[
F_1 = \frac{1}{N} \sum_{i=1}^{N} \frac{2 \|\hat{y}_i \cap y_i\|}{\|\hat{y}_i\| + \|y_i\|} \tag{29}
\]

where \(N\), \(y_i\), and \(\hat{y}_i\) are the same as precision calculation formula.

**Accuracy:** accuracy evaluates Jaccard similarity between the ground truth labels and the predicted labels

\[
Acc = \frac{1}{N} \sum_{i=1}^{N} \frac{\|\hat{y}_i \cap y_i\|}{\|\hat{y}_i \cup y_i\|} \tag{30}
\]

where \(N\), \(y_i\), and \(\hat{y}_i\) are the same as precision calculation formula.

**Hamming loss:** Hamming loss evaluates the fraction of misclassified instance-label pairs, where a relevant label is missed or an irrelevant is predicted,

\[
HL = \frac{1}{N} \sum_{i=1}^{N} \sum_{l=1}^{L} \frac{1}{\|\hat{y}_i\| + \|y_i\|} \tag{31}
\]

where \(\triangle\) is the element difference between two sets, \(L\) is the size of label space.

C. EXPERIMENT SETTING
In this paper, the PyTorch 1.7.1 is used to build a network model and experiments are carried out on a NVIDIA GeForce GTX 3090 for Ubuntu 18.04 LTS system. The proposed model use BERT-wwm model as our language model to get semantic representation vector of event text and label text, in addition the GCN is used to extract event topic information [5]. The detailed hyper parameters are shown in Table 2.

| parameter          | value  |
|--------------------|--------|
| LM embedding dim   | 768    |
| LM vocabulary size | 21128  |
| LM sequence length | 512    |
| LM hidden size     | 768    |
| LM hidden layers   | 12     |
| LM attention heads | 12     |
| GCN embedding dim  | 768    |
| Optimizer          | Adam   |
| Learning rate      | 10e-5  |
| Learning rate decay| 0.5    |
| Batch size         | 16     |

D. BASELINES
To evaluate the multi-label classification model, compare the proposed method with the following state-of-the-art baseline methods on constructed datasets.

**SVM with TF-IDF:** This paper use frequency–inverse document frequency (TF-IDF) to construct features and train a separate binary classifier for each label. This paper compares it with many machines learning algorithms, including support vector machine (SVM), logistic regression, random forest, etc. The experiment results show that SVM with TF-IDF model obtains better performance in constructed datasets than other models.
CNN [36]: uses multiple convolution kernels to extract text features, which are then inputted to the linear transformation layer followed by a sigmoid function to output the probability distribution over the label space. The multi-label soft margin loss is optimized.

CNN-RNN [37]: utilizes CNN and RNN to capture both the global and local textual semantics and model the label correlations.

ML-Net [38]: an end-to-end deep learning framework for multi-label classification. ML-Net combines a label prediction network with an automated label count prediction mechanism to output an optimal set of labels by leveraging both predicted confidence score of each label and the contextual information in the target document.

SGM [11]: it takes the multi-label classification task as a sequence generation problem. To address it, SGM applies a sequence generation model with a decoder structure. It not only captures the correlations between labels, but also selects the most informative words automatically when predicting different labels.

HMC-CLC [16]: A hierarchical multi-label classification method with class label correlation (HMC-CLC) which exploits the label correlations of different branches to benefit the discrimination of hierarchical multilabel classification. For each label in the hierarchy, HMC-CLC uses feature incremental learning to encode the labels of different branches into the input space, reflecting the label correlations of different branches by the weights of classification model in corresponding dimensions. In addition, a greedy label selection method is used to dynamically decide the correlated labels of different branches for each label.

GUDN [40]: A novel guide network including two modules of guide and two loss functions to instruct the BERT on extracting features, meanwhile GUDN also combines raw label semantics and pre-trained model to find the latent space between texts and labels.

MrMP [17]: A Transformer based message passing network for the multi-label text classification problem considering two types of statistical relations between labels, and designed a multi-relation label embedding module to account for the dependencies between labels.

### E. RESULTS AND ANALYSES

This paper divides the constructed dataset into training set, validation set and test set, and experiment on the test set with models including our proposed methods and several baselines. The results of the experiment are shown in Table 3. Consider from the hamming-loss and F1 scores of multiple machine learning methods, SVM is the best model, while it still has 12.79% gap of loss and 2.33% gap of micro-F1 comparing with our proposed methods. This result shows that the effect of our methods has significantly improved compared with traditional methods based on machine learning. This improvement can be explained by the difficulty of extracting accurate semantic information from text for machine learning methods. Pre-trained language models with rich semantic information can effectively bridge this gap, which achieves the optimal model effect.

The proposed model also has the best performance on the main evaluation metrics compared to other deep learning baselines. For example, the proposed model achieves 15.73% reduction in hamming-loss and 2.69% improvement in micro-F1 score compared with the best baseline result. This is because the proposed model not only obtains contextual information through BERT, but also obtains topic information contained in the text through GCN, which provides richer semantic information for the model and makes the prediction result more accurate.

### TABLE 3. Comparison result.

| Models   | Acc(+) | P(+) | R(+) | F1(+) | HL(-) |
|----------|--------|------|------|-------|-------|
| SVM      | 0.3438 | 0.4013 | 0.4096 | 0.4054 | 0.066873 |
| CNN      | 0.4454 | 0.5189 | 0.5293 | 0.5240 | 0.005599 |
| CNN-RNN  | 0.4467 | 0.5181 | 0.5299 | 0.5239 | 0.005508 |
| ML-Net   | 0.4485 | 0.5193 | 0.5313 | 0.5252 | 0.005494 |
| SGM      | 0.4528 | 0.5229 | 0.5339 | 0.5283 | 0.005453 |
| HMC-CLC  | 0.4600 | 0.5287 | 0.5395 | 0.5340 | 0.005385 |
| GUDN     | 0.4627 | 0.5306 | 0.5415 | 0.5360 | 0.005363 |
| MrMP     | 0.4623 | 0.5303 | 0.5407 | 0.5355 | 0.005366 |
| proposed model | 0.4782 | 0.5428 | 0.5531 | 0.5479 | 0.005221 |
TABLE 4. Ablation of label count prediction module.

| Models       | Acc(+) | P(+)   | R(+) | F1(+) | HL(-)   |
|--------------|--------|--------|------|-------|---------|
| without GCN | 0.3510 | 0.4117 | 0.4209 | 0.4163 | 0.006752 |
| without BERT | 0.3345 | 0.3916 | 0.4002 | 0.3958 | 0.006988 |
| proposed model | 0.4782 | 0.5428 | 0.5531 | 0.5479 | 0.005221 |

TABLE 5. Ablation of event label semantic insert module.

| Models      | Acc(+) | P(+)   | R(+) | F1(+) | HL(-)   |
|-------------|--------|--------|------|-------|---------|
| without MN  | 0.4130 | 0.4802 | 0.4913 | 0.4857 | 0.005951 |
| proposed model | 0.4782 | 0.5428 | 0.5531 | 0.5479 | 0.005221 |

F. ABLATION ANALYSIS

To validate the effectiveness of event label count prediction module and event label semantic insert module in the proposed method, this paper conduct ablation experiments on these modules respectively.

1) LABEL COUNT PREDICTION MODULE
Label count prediction module has two feature extractors: GCN and BERT. In order to prove the effectiveness of these two feature extractors, this paper carried out ablation experiments on them.

Ablation experiment of GCN: GCN was removed, only BERT was used as features extractor, other parts remained unchanged.

Ablation experiment of BERT: BERT was removed, only GCN was used as features extractor, other parts remained unchanged.

Table 4 shows the ablation experiment results of the label count prediction module, where “without GCN” is the results of the ablation GCN and “without BERT” is the results of the ablation BERT. It can be seen that our method outperforms on both ablation methods in all evaluation metrics by more than 3% on average.

For the model with GCN ablation, the experimental results are inferior to the model without ablation. An explainable reason is that the “without GCN” model cannot extract global topic information efficiently and accurately due to the lack of GCN’s ability of encoding topic information. As for the model that ablates pre-trained language model, it is difficult to extract syntactic structure information of longer texts due to the complexity of long text, so the results of the “without BERT” model are not as good as those of the model with pretrained language model. The results of the ablation experiments show the effectiveness of the proposed event text classification method.

2) EVENT LABEL SEMANTIC INSERT

The ablation of event label semantic insert module is the ablation of memory network, only use full connected layer and softmax to get the classification result.

As can be seen from the values in Table 5, the proposed method has an improvement of about 5% in accuracy and about 6% in micro-F1 score compared with the “without MN” model, which is significant in terms of effect. Considering the memory and comparison process of the memory network, the proposed method can make full use of the semantic information contained in the text and the label text information. At the same time, the multi-layer Transformer structure inspired by BERT allows the model to obtain surfaces feature, syntactic features and semantic features from label texts, which leads to the improvement of accuracy in identifying different labels.

V. CONCLUSION

Aiming at the features of Chinese government hotline texts, a multi-label classification framework for the event text based on GCN-BERT and memory is proposed in this paper. The proposed framework consists of a label count prediction module, a label semantic insert module and a label selection module. In label count prediction module, this paper use GCN and BERT to extract the event topic information and semantic information respectively and fuse them into a fusion vector to predict the label count. In label semantic insert module, this paper takes advantage of the answer selection framework to obtain the event label candidate set with the help of a multi-hop memory network. In label selection module, the proposed model sorts the event label candidate set and generate the optimal multiple labels of the event. The innovations lie in employing AMR-seq2seq based graph constructing algorithm to process the event text and obtain the event graph and use the semantic information of the Chinese government hotline text to predict the label count to guide more accurate label
generation and improve the adaptability of label prediction. This paper contribute to provide an alternative framework for data processing of the Chinese government hotline. In the future work, this paper will study a better method to construct event graph for more sufficient extraction of relevant information between text sentences than the AMR seq2seq algorithm.

REFERENCES

[1] X. She, J. Chen, and G. Chen, “Joint learning with BERT-GCN and multi-attention for event text classification and event assignment,” IEEE Access, vol. 10, pp. 27031–27040, 2022, doi: 10.1109/ACCESS.2022.3156918.

[2] X. Peng, Y. Liang, and L. Y. Xu, “An approach for discovering urban public management problem and optimizing urban governance based on 12345 citizen service hotline,” Acta Scientiarum Naturalium Universitatis Pekinensis, vol. 56, no. 4, pp. 721–731, 2020.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in Proc. ICLR, Scottsdale, AZ, USA, 2013, pp. 1–12.

[4] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in Proc. EMNLP, Doha, Qatar, 2014, pp. 1532–1543.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.

[6] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, and R. Salakhutdinov, “XLNet: Generalized autoregressive pretraining for language understanding,” in Proc. 51th NeurIPS, Vancouver, BC, Canada, 2019, pp. 5754–5764.

[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” 2019, arXiv:1907.11692.

[8] L. Banarsec, C. Bonial, S. Cai, M. Georgescu, K. Griffith, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, “Abstract meaning representation for sembanking,” in Proc. LAW-IF, 2013, pp. 178–186.

[9] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2016, arXiv:1609.02907.

[10] J. Weston, S. Chopra, and A. Bordes, “Memory networks,” 2014, arXiv:1410.3916.

[11] T. Mikolov, H. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in Proc. ICLR, Scottsdale, AZ, USA, 2013, pp. 1–12.

[12] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in Proc. EMNLP, Doha, Qatar, 2014, pp. 1532–1543.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.

[14] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, and R. Salakhutdinov, “XLNet: Generalized autoregressive pretraining for language understanding,” in Proc. 51th NeurIPS, Vancouver, BC, Canada, 2019, pp. 5754–5764.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” 2019, arXiv:1907.11692.

[16] L. Banarsec, C. Bonial, S. Cai, M. Georgescu, K. Griffith, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, “Abstract meaning representation for sembanking,” in Proc. LAW-IF, 2013, pp. 178–186.

[17] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2016, arXiv:1609.02907.

[18] J. Weston, S. Chopra, and A. Bordes, “Memory networks,” 2014, arXiv:1410.3916.

[19] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, “SGM: Sequence generation model for multi-label classification,” 2018, arXiv:1806.04822.

[20] J. Lin, Q. Su, P. Yang, S. Ma, and X. Sun, “Semantic-unit-based dilated convolution for multi-label text classification,” 2018, arXiv:1808.05601.

[21] J. Lanchantin, A. Sekhon, and Y. Qi, “Neural message passing for multi-label classification,” in Machine Learning and Knowledge Discovery in Databases. Würzburg, Germany: Springer, Sep. 2019, pp. 138–163.

[22] W. Hong, W. Xu, J. Qi, and Y. Weng, “Neural tensor network for multi-label classification,” IEEE Access, vol. 7, pp. 96936–96941, 2019.

[23] S. Wang, G. Peng, and Z. Zheng, “Capturing joint label distribution for multi-label classification through adversarial learning,” IEEE Trans. Knowl. Data Eng., vol. 14, no. 12, pp. 2310–2321, Dec. 2020.

[24] Z. Xu, B. Zhang, D. Li, and X. Yue, “Hierarchical multilabel classification by exploiting label correlations,” Int. J. Mach. Learn. Cybern., vol. 13, pp. 115–131, Sep. 2021.

[25] M. Ozmen, H. Zhang, P. Wang, and M. Coates, “Multi-relation message passing for multi-label text classification,” 2022, arXiv:2202.04844.

[26] L. Yao, C. Mao, and Y. Luo, “Graph convolutional networks for text classification,” 2018, arXiv:1809.05679.

[27] F. Lei, X. Liu, Z. Li, Q. Dai, and S. Wang, “Multihop neighbor information fusion graph convolutional network for text classification,” Math. Problems Eng., vol. 2021, pp. 1–9, Apr. 2021.

[28] H. Tang, Y. Mi, F. Xue, and Y. Cao, “An integration model based on graph convolutional network for text classification,” IEEE Access, vol. 8, pp. 148865–148876, 2020.

[29] L. Huang, D. Ma, S. Li, X. Zhang, and H. Wang, “Text level graph neural network for text classification,” 2019, arXiv:1910.02356.

[30] K. Zhao, L. Huang, R. Song, Q. Shen, and H. Xu, “A sequential graph neural network for short text classification,” Algorithms, vol. 14, no. 12, pp. 352–360, 2021.

[31] 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.