A Hierarchal BERT Structure for Native Speaker Writing Detection

Xinyuan Wang
Systems Engineering Institute
Xi'an Jiaotong University
Xi'an, China
wxy0713@stu.xjtu.edu.cn

Qinke Peng*
Systems Engineering Institute
Xi'an Jiaotong University
Xi'an, China
qkpeng@xjtu.edu.cn

Xu Mou
Systems Engineering Institute
Xi'an Jiaotong University
Xi'an, China
muxu19950916@xjtu.edu.cn

Haozhou Li
Systems Engineering Institute
Xi'an Jiaotong University
Xi'an, China
lihaozhou1126@stu.xjtu.edu.cn

Ying Wang
Systems Engineering Institute
Xi'an Jiaotong University
Xi'an, China
ying_wang@xjtu.edu.cn

***Abstract*—Native speaker detection has always focused on speech data. However, the writing style of native speakers is also different from that of non-native speakers. Therefore, for the first time, we performed native speaker detection on text data so that non-native speakers can better learn the writing style of native speakers. Native speaker writing detection is relatively difficult due to the long sequences and complex semantics in the writings. Therefore, we use BERT-based methods. However, BERT suffers from the exponentially increasing computational complexity because of the self-attention mechanism, which limits the length of text input. Consequently, in this paper, we present a hierarchical BERT model to solve this problem. Our model first cuts the long text into segments and obtains segment representation vectors from BERT. Then, we extract the temporal and interactional information between segments to form a text-level representation vector for writing detection. We conducted experiments on a self-made native speaker writing detection dataset. The results demonstrate that our model can accurately recognize native speakers' writing. In addition, we have successfully used it in various long text classification tasks and achieved improvement over the baseline models. We also show the importance of both temporal and interaction information for text-level representation.**

***Index Terms*—Native Speaker Writing Detection, BERT, Transformer, Long Text Classification, Hierarchical Structure**

I. Introduction

There are corresponding differences between native speakers and non-native speakers in listening, speaking, reading, and writing [1] [2]. Many native speaker detection methods have been proposed to distinguish whether these people are native speakers [3] [4]. However, today's methods focus on speech recognition, which uses metrics like prosody, pauses, and coherence to tell if a speaker is a Native speaker. These features no longer exist in texts. Only through the arrangement and combination of words can texts be distinguished. Thersfore, our goal is to perform a BERT-based hierarchal structure and determine whether the author of an article is a native English speaker. As we all know, there are millions of words in English, which bring countless combinations of those words. Nevertheless, not every combination makes sense. Using the right combinations is the key point to correct presentation. So learning a language is to learn the rules of those combinations, what we call grammar. In order to identify whether a text is grammatical and written by a native speaker, word combination and order in the text are very important. Regarding native speaker writing detection, the lack of a dataset prompts us to form a new dataset based on scientific papers. The dataset we collected is based on scientific papers on the web of science (WOS). The texts for classification are drawn from the abstract of the papers. And then, we labeled them according to the authors of each paper.

The knowledge learned from native speaker detection can be transferred to later works, such as language polish and auto dialogue. Native speaker writing detection is relatively difficult due to the complex semantics and long sequences in the writings.

Bidirectional Encoder Representations from Transformers (BERT) [5] is a fancy pre-trained language model based on self-attention [6] for representing the text, which can extract very information from the combinations between tokens. It is a text representation perfectly suited to Native speaker writing detection. Native speakers are those who can give valid judgments on their language and identify ill-formed grammatical expressions in their languages [7], and BERT is also capable of doing this task.

Hierarchical methods [8] have been proposed to reduce the computational complexity. However, at the same time, some new questions emerged, such as how to divide and reconstruct articles. The first question can be solved using sliding windows, and the second has a solution of segment-level attention. Furthermore, the methods of recurrence network over BERT and Transformer over BERT have been published to do the reconstruction [9]. However, even though these

* Corresponding Author.

methods have made considerable advancements, they do not engage all matters of thinking. The segment interaction and temporal information are essential in the reconstruction task. Consequently, this motivates us to employ both segment-level attention and time series analysis methods in this task.

In this paper, we design a new end-to-end BERT-based hierarchical deep learning framework for text classification, which considers segment interaction and temporal information. Due to the hierarchical structure, two levels of information are extracted from texts: word-level and segment-level. First, we split the long text input into several shorter segments through sliding windows and then build the representation of segments using self-attention to capture inter words information in each segment. Then, we aggregate those segment representations into a document representation. In order to make good use of both the interactional and temporal information, the LSTM [10] network and GAT [11] are employed successively to explore links between segment representations. By adding these two sections, we call this architecture BERT-LSTM-GAT. Finally, we experiment on the standard and well-used datasets and our native speaker detection dataset and then verify the performance.

Our contributions are:

- We propose an extension to the BERT model based on the hierarchical structure, which considers both the interactional and temporal information.
- We collect a new native speaker writing detection dataset based on scientific papers' abstracts.
- We have achieved better performance on the native speaker writing detection dataset and other text classification datasets.

II. Related Work

A. Native Speaker Detection

Speech signals are the most natural and popular means of communication between people. Speech recognition refers to the recognition and classification of the input speech signal to get the information we need. Many studies have focused on dividing speakers into native and non-native speakers based on phonetic information. An ASR can be divided into four parts: preprocessing part, feature extraction part, classification model, and language model. The two most essential parts are feature extraction and classification model. The feature extraction part is to extract the unique information in the speech signal except for the text content by some mathematical means. The classification model uses the features extracted in the previous stage as input to predict and classify the text. Compared with the text-based classification, speech information has not only text features, but also much characteristic information. Most of these features express the emotional information in speech. Speech features can be divided into four categories [12]: continuous feature, qualitative feature, spectral feature, and TEO (Teager energy operator) [13] based feature. Various feature extraction techniques based on mathematical methods [14] have been developed to extract the above features from speech signals.

B. Bert Based Classification Models

When the attention mechanism became widely used, there were numerous attention types. The most famous one must be the self-attention-based Transformers [6]. On the fundamental of the Transformer, many pre-trained language models have arisen. These pre-trained models obtain task-independent model parameters from large-scale data through self-supervised learning and apply them to various downstream tasks, including text classification. Even though they are not specially designed for text classification, their strong characterization ability, wide application range, and excellent generalized ability guarantee excellent performance. Since their publication, they have quickly become one of the most famous lines of research worldwide, and some researchers have even applied the concept to other fields besides NLP [15] [16].

III. Proposed Framework Structure

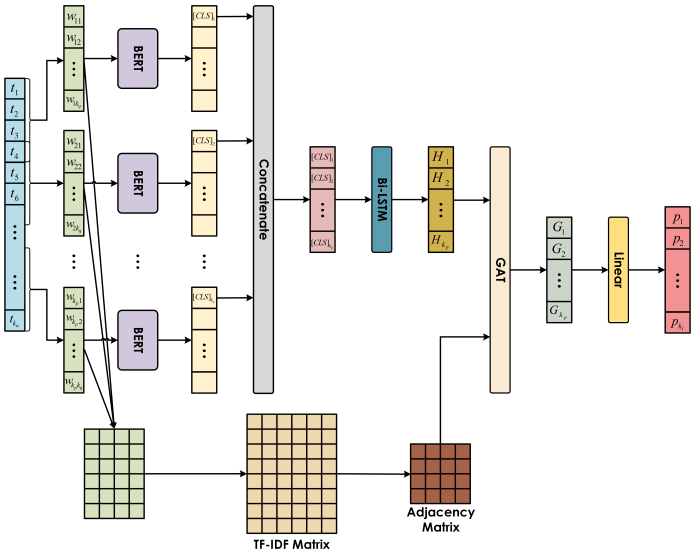

Fig. 1. Structure of the proposed hierarchical framework.

To extract potential information from the text to determine whether native speakers wrote the text. Our model uses hierarchical structure and subsequent combinatorial networks to extract temporal and combinatorial information from the text. Our network framework is shown in Fig. 1. Our model includes two attention mechanisms—one at the word level and one at the sentence level—that let the model pay more or less attention to individual words and sentences when constructing the document representation. Firstly, the texts are divided into several pieces, and the relationships between pieces are measured by the cosine similarity between their TF-IDF [17] vectors. Then, the BERT is employed to obtain the representation of each segment. After that, we use LSTM and GAT to extract the interactional and temporal information successively. Finally, we obtain the classification distribution. In the following of this section, we will describe each part of the model in detail.

A. Problem Definition

TABLE I
NOTATIONS OF FREQUENTLY USED VARIABLES

Notations	Descriptions
T	The initial long text document
t_i	The i th word of T
k_w	The number of total words of T
s_i	The i th segment of T
k_p	The number of segment in a document
w_{ij}	The j th word of segment s_i
k_q	The number of words in one segment
$\boldsymbol{C}_l$	The real label distribution
$\boldsymbol{C}_p$	The predicted classification distribution
p_i	The possibility of the text to be the i th category
k_l	The number of total categories
k_m	The number of tokens in one segment
k_h	The length of every token embedding vector
L	The number of Transformer Encoder layers
k_{mhB}	The number of attention heads in BERT
k_{mhG}	The number of attention heads in GAT

Every long text data can be defined as a set of many words, as shown as $T = \{t_1, t_2, \ldots, t_{k_w}\}$, where t_i is the ith words in the text. The hierarchical structure requires the segment, which results in the representation of text $T = \{s_1, s_2, \ldots, s_{k_p}\}$. And each segment $s_i \in T, i \in [1, k_p]$ contains k_q words $w_{i1}, w_{i2}, \ldots, w_{ik_q}$ so the theoretical maximal word number representing a long text is $k_p \cdot k_q$. However, the sliding window is used to make the segment, which means $k_w < k_p \cdot k_q$. The actual distribution of the text label is $\boldsymbol{C}_l = \{c_1, c_2, \ldots, c_{k_l}\}$. c_1 is 1 if the text is of class i, and 0 otherwise. Our task is to predict its classification distribution $\boldsymbol{C}_p = \{p_1, p_2, \ldots, p_{k_l}\}$ and then find the possible category to be the final label. The summary of variable symbols frequently used in this article is shown in TABLE I.

B. Word Wise Self-Attention: Bert Part

In this part, the long text is segmented into pieces. The BERT representation of each segment and the adjacent matrix among segments are generated. The `{cls}` token summarizes the information in segments.

As we just defined, there are k_w words and k_p segments in the given long text:

$$T = \{t_1, t_2, \ldots, t_{k_w}\} = \{s_1, s_2, \ldots, s_{k_p}\} \tag{1}$$

For every segment, the number of words in one segment is k_q. However, not each text has the same length, so the padding is applied to meet the requirement of k_q.

$$\boldsymbol{X}_i^{emb} = \mathrm{BERTemb}\left(s_i\right) \tag{2}$$

Where $\boldsymbol{X}_i^{emb} \in \mathbb{R}^{k_m \times k_h}$ is the BERT vectorization matrix of the text segment s_i. $k_m \geq k_q + 2$ is the number of tokens generated by k_q words. Because of the `{cls}` and `{sep}` token , k_m is required to be equal to or larger than $k_q + 2$. And k_h means the length of every token embedding vector, which is also the hidden size of the BERT. For the segment embedding $\boldsymbol{X}_i^{emb}$, an L layer Transformer Encoder is used to generate the final segment representation by using the bidirectional self-attention, and the interaction among tokens is learned to form the new representation matrix $\boldsymbol{X}_i^L$. First, we assign $\boldsymbol{X}_i^0 = \boldsymbol{X}_i^{emb}$, and then for every Transformer Encoder layer, output $\boldsymbol{X}_i^l$ is generated from input $\boldsymbol{X}_i^{l-1}$ by following the multi-head self-attention mechanism:

$$\begin{aligned}
\boldsymbol{Q}_i^{l,j} &= \boldsymbol{X}_i^{l-1} * \boldsymbol{W}\boldsymbol{Q}_i^{l,j}, \boldsymbol{Q}_i^{l,j} \in \mathbb{R}^{k_m \times k_h} \\
\boldsymbol{K}_i^{l,j} &= \boldsymbol{X}_i^{l-1} * \boldsymbol{W}\boldsymbol{K}_i^{l,j}, \boldsymbol{K}_i^{l,j} \in \mathbb{R}^{k_m \times k_h} \\
\boldsymbol{V}_i^{l,j} &= \boldsymbol{X}_i^{l-1} * \boldsymbol{W}\boldsymbol{V}_i^{l,j}, \boldsymbol{V}_i^{l,j} \in \mathbb{R}^{k_m \times k_h}
\end{aligned} \tag{3}$$

Here, $\boldsymbol{Q}_i^{l,j}$, $\boldsymbol{K}_i^{l,j}$, $\boldsymbol{V}_i^{l,j}$ stand for query, key, and value vectors in the i th segment of the l th layer. Then the self-attention was calculated as:

$$\boldsymbol{S}_i^{l,j} = \mathrm{Softmax}\left(\frac{\boldsymbol{Q}_i^{l,j} * \boldsymbol{K}_i^{l,j^{\mathrm{T}}}}{\sqrt{k_h}}\right) * \boldsymbol{V}_i^{l,j}, \boldsymbol{S}_i^{l,j} \in \mathbb{R}^{k_m \times k_h} \tag{4}$$

Where $S_i^{l,j}$ means the jth head attention, then multiplying by $\boldsymbol{V}_i^{l,j}$ gives the embedding obtained by inter-word attention weighting. Then concatenate k_{mhB} head attentions and obtain multi-head attention representation:

$$\boldsymbol{S}_{i(1)}^l = \left[\boldsymbol{S}_i^{l,1}, \ldots, \boldsymbol{S}_i^{l,j}, \ldots\right] \tag{5}$$

$$\boldsymbol{S}_{i(2)}^l = \boldsymbol{S}_{(1)}^l * \boldsymbol{W}\boldsymbol{O}_i^l, \boldsymbol{S}_{i(2)}^l \in \mathbb{R}^{k_m \times k_h} \tag{6}$$

Then feedforward layers, residual connections, and layer normalization are performed to make the model converge.

$$\boldsymbol{O}_{i(1)}^l = \mathrm{LayerNorm}\left(\boldsymbol{X}_i^{l-1} + \boldsymbol{S}_{i(2)}^l\right), \boldsymbol{o}_{i(1)}^l \in \mathbb{R}^{k_m \times k_h} \tag{7}$$

$$\boldsymbol{S}_{i(3)}^l = \mathrm{Relu}\left(\boldsymbol{O}_{i(1)}^l * \boldsymbol{W}_i^l + \boldsymbol{b}\boldsymbol{F}_i^l\right), \boldsymbol{S}_{i(3)}^l \in \mathbb{R}^{k_m \times k_h} \tag{8}$$

$$\boldsymbol{X}_i^l = \mathrm{LayerNorm}\left(\boldsymbol{O}_{i(1)}^l + \boldsymbol{S}_{i(3)}^l\right), \boldsymbol{X}_i^l \in \mathbb{R}^{k_m \times k_h} \tag{9}$$

The final segment representation matrix is shown as $\boldsymbol{X}_i^L$, where $\boldsymbol{X}_i^L \in \mathbb{R}^{k_m \times k_h}$. The final segment representation vector $\boldsymbol{P}_i$ is the vector of `{cls}` token in $\boldsymbol{X}_i^L$, and $\boldsymbol{P}_i \in \mathbb{R}^{k_h}$. There are $\boldsymbol{k}_p$ segments in one text, so we combine these vectors into a matrix $\boldsymbol{P}_B$

$$\boldsymbol{P}_B = \left[\boldsymbol{P}_1, \ldots, \boldsymbol{P}_i, \ldots\right] \tag{10}$$

Here the matrix $\boldsymbol{P}_B$ denotes the BERT representation of the long text based on the segmentation.

C. Temporal Information Extraction: LSTM Part

After obtaining the block representation $\boldsymbol{P}_B$ from the last section, the Bi-LSTM is used to learn the temporal information among segments to obtain article representations. For every segment, the temporal information is extracted as:

$$\overrightarrow{\boldsymbol{H}}_i = \overrightarrow{\mathrm{LSTM}}\left(\boldsymbol{P}_i\right), i \in [1, k_p] \tag{11}$$

$$\overleftarrow{\boldsymbol{H}}_i = \overleftarrow{\mathrm{LSTM}}\left(\boldsymbol{P}_i\right), i \in [k_p, 1] \tag{12}$$

$$\boldsymbol{H}_i = \left[\overrightarrow{\boldsymbol{H}}_i, \overleftarrow{\boldsymbol{H}}_i\right], \boldsymbol{H}_i \in \mathbb{R}^{2 \cdot k_h} \tag{13}$$

$\boldsymbol{H}_i$ represents the ith segment after LSTM processing, while $\overrightarrow{\boldsymbol{H}}_i$ and $\overleftarrow{\boldsymbol{H}}_i$ represent the forward LSTM representation and backward LSTM representation, respectively. Then, we obtain an annotation of the ith segment representation $\boldsymbol{H}_i$ by concatenating $\overrightarrow{\boldsymbol{H}}_i$ and $\overleftarrow{\boldsymbol{H}}_i$. $\boldsymbol{H}_i$ represents the neighboring segments around block $\boldsymbol{H}_i$ but still focuses on segment i. Hence, we can get the segment representations containing the sequence information among segments.

D. Segment Interaction: GAT Part

Apart from the temporal information, the interactions among segments are similarly crucial. Unlike the regular method that employs the self-attention mechanism to measure the importance of segment features, we introduce the graph attention network (GAT), which can reward the segments crucial for predicting the final classification via graph data structure. The graph data structure consists of vertexes and edges. Vertexes represent entities and edges represent relationships between two entities. The relationship here is expressed as connected or not, mathematically expressed as $\mathbf{0}$ or $\mathbf{1}$. In GAT, the attention weights between vertexes are first learned, then, according to the edges and learned weights, the new presentations are calculated from the former presentations. Only the weights between connected vertexes are recalculated.

In this part, the segment representations are regarded as vertexes of graph structure, while the connections between segments are seen as edges. Two vertexes are not considered connected until their similarity exceeds a certain threshold. The similarity here is measured by the Euclidean distance between TF-IDF vectors of each segment.

First, for a pair of connected vertexes $(\boldsymbol{H}_i, \boldsymbol{H}_j)$, known as the representations of two different segments, the pair-wise unnormalized attention coefficient between two representations was computed to describe the relationship e_{ij}.

$$\begin{gathered} e_{ij} = \mathrm{LeakyRelu}\left(a^T\left(\boldsymbol{W} * \boldsymbol{H}_i \| \boldsymbol{W} * \boldsymbol{H}_j\right)\right) \\ e_{ij} \in \mathbb{R} \\ i \in [1, k_p] \end{gathered} \tag{14}$$

$\boldsymbol{W} \in \mathbb{R}^{2 \cdot k_h \times k_h}$ is a learnable projection matrix, while a is a $\mathbb{R}^{k_h} \times \mathbb{R}^{k_h} \rightarrow \mathbb{R}$ map to compute attention coefficients. In contrast with the dot-product attention used for the Transformer model, attention here is usually called additive attention. Additionally, $\|$ here denotes concatenation, which makes $e_{ij} \neq e_{ji}$. In other words, attention is asymmetrical.

In addition to the concatenation, the attention of all neighbors of each vertex needs to be normalized when the neighbor information is aggregated. The normalized attention weight α_{ij} is the actual aggregation coefficient.

$$\alpha_{ij} = \mathrm{Softmax}\left(e_{ij}\right) = \frac{e^{ij}}{\sum_{k \in k_p} e^{e_{ik}}} \tag{15}$$

The normalization here further leads to asymmetries in the weight of attention. Because in the normalization process, the normalized object of each vertex is different. e_{ij} is normalized for all neighbors of the vertex i, while e_{ji} is normalized for all neighbors of the vertex j. So α_{ij} here is also asymmetrical. The asymmetry of attention weight further liberates the attention mechanism so that the interaction between two vertexes can differ.

Finally, the representations of neighbors are aggregated together, scaled by the attention scores. What's more, we use multi-head attention to enrich the model capacity and stabilize the learning process, where k_{mhG} means the head number and $\boldsymbol{W}^{(k)}$ means the learnable matrix in the k th head.

$$\begin{gathered} \boldsymbol{G}_i = \sigma\left(\frac{1}{k_{mhG}} \sum_{k=1}^{k_{mhG}} \sum_{j \in k_p} \alpha_{ij}^{(k)} \cdot \left(\boldsymbol{W}^{(k)} \boldsymbol{H}_j\right)\right) \\ \boldsymbol{G}_i \in \mathbb{R}^{k_h} \\ i \in [1, k_p] \end{gathered} \tag{16}$$

E. Final Prediction Part

We put the sum of all segment representations into a dense layer for a final passage classification to make the final decision.

$$\begin{gathered} \boldsymbol{C}_p = \mathrm{Softmax}\left(\left(\sum_{i=1}^{k_p} \boldsymbol{G}_i\right) * \boldsymbol{W}_F + \boldsymbol{b}_F\right) \\ \boldsymbol{W}_F \in \mathbb{R}^{k_h \times k_l}, \boldsymbol{b}_F \in \mathbb{R}^{k_l} \end{gathered} \tag{17}$$

$\boldsymbol{C}_p = \{p_1, p_2, \ldots, p_{k_l}\}$ here means the predicted classification distribution. Compared with the actual distribution $\boldsymbol{C}_l = \{c_1, c_2, \ldots, c_{k_l}\}$. c_1, the average cross-entropy loss was used as the loss function in training to measure the difference between the target distribution $\boldsymbol{C}_l$ and the predicted distribution $\boldsymbol{C}_p$.

$$\mathrm{loss} = -\sum_{i=1}^{k_l} c_i \log\left(p_i\right) \tag{18}$$

IV. Experiments and Analysis

To verify the performance of the proposed structure, we evaluated our models on several different datasets. These datasets contain different task types, including sentiment analysis, topic labeling, and news classification.

A. Dataset and statistic

We propose a new dataset based on papers in the Web of Science (WOS). We expect to collect a dataset for native speaker detection. To achieve this goal, we downloaded more than 15,000 papers. Next, we cleaned the data and labeled them according to the authors' last names. We preliminarily classified the data into the Sino-Tibetan language family and

Indo-European language family, which is a binary dataset. Finally, there are 14477 abstracts. We applied this brand-new dataset to the experiment to determine whether an English paragraph was written by a native speaker. Despite the native speaker detection datasets, we also use some standard and well-used datasets to test our structure.

Table II shows the data Statistic. It can be seen from Table II that all datasets except IMDB and Native Speaker datasets correspond to multi-classification tasks.

TABLE II
RESULT OF THE DATA STATISTIC

Dataset	Class	Number	Average Token	Longest
Native Speaker	2	14641	565.17	1936
IMDB Reviews	2	50000	290.35	2470
20 newsgroups	20	18846	409.10	20235
AG news	4	127600	31.59	194
WOS 11967	7	11967	271.38	1102

B. Baseline model

To verify the performance of the proposed model, we have done the classification using several general classification models and Bert-based models,including RoBERT and ToBERT. We also use neural networks based on RNN and CNN, including some variants and networks with added attention mechanisms.

- RoBERT: means recurrence over BERT.
- ToBERT: means Transformer over BERT. Compared to the recurrence network, Transformer effectively captures the long-distance relationship between words in a text sequence.

C. Results and analysis

Table III shows the performance of each model in the native speaker detection task.

TABLE III
RESULT ON NATIVE SPEAKER DETECTION TASK

Models	Accuracy	F1-Score	Precision	Recall
LSTM	71.59%	59.80%	66.91%	54.06%
Self-Attention	78.08%	70.21%	74.90%	66.08%
LSTM-Attention	77.07%	68.40%	74.14%	63.49%
RNN	65.98%	52.23%	57.88%	47.59%
CNN	76.38%	65.17%	76.92%	56.54%
RCNN	79.10%	71.45%	76.65%	66.90%
RoBERTa	82.27%	81.14%	82.07%	80.90%
BERT mean	81.68%	76.20%	77.40%	75.03%
RoBERT	82.55%	77.43%	78.31%	76.56%
ToBERT	76.84%	69.86%	71.10%	68.67%
BERT LSTM GAT	83.24%	81.31%	84.90%	80.04%

In the BERT-LSTM-GAT network, the BERT model can obtain segment representations, and the LSTM network represents the sequential relationship between segments, similar to the position embedding in the Transformer input embedding layer. Then, the correlation between segments is described according to TF-IDF vectors, and the segment representation vectors are updated with the GAT network to describe the interaction between segments.

We have shown that BERT-LSTM-GAT gets the final prediction of a long document based on a simple baseline of the average split prediction. Our results confirm that BERT-LSTM-GAT can be used for long sequences and has competitive performance and lower computational complexity. Results that use BERT-LSTM-GAT are better than baseline models, including RoBERT and ToBERT.

It can be seen that our model has superior performance compared to the typical neural network classifiers. Both the large number of parameters of BERT and the feature extraction of the hierarchical structure have made extraordinary contributions.

Table IV presents the overall results of comparative tests performed on multiple datasets. The evaluation index adopted here is accuracy.

TABLE IV
RESULT ON OTHER CLASSIFICATION TASK

Models	IMDB	20 News	AG news	WOS
BERT mean	92.41%	81.39%	92.76%	92.27%
RoBERT	93.82%	80.52%	92.86%	93.23%
ToBERT	93.92%	84.88%	93.30%	94.15%
BERT LSTM GAT	94.11%	84.82%	93.42%	94.21%

As can be seen from Table IV, the BERT-LSTM-GAT method has achieved good results on these data sets. The best results were achieved on all datasets except 20NewsGroup. In the 20Newsgroup dataset, comparing the effects of Bert-Mean and Bert-LSTM models, the performance of the model with the LSTM network is decreased, but models with Transformer and GAT have an excellent complement. This method improves the accuracy and efficiency of the model and can be widely applied to data types and application fields.

D. Ablation Experiment

To verify the importance of the LSTM network to the overall structure, we set up the BERT-GAT network. Furthermore, we use the basic version of BERT to classify the dataset to explore the effect of hierarchical structure and temporal information extraction on the model performance.

Table V shows the results of ablation experiments in the native speaker detection task. The result indicates that both hierarchical structure and recurrent part contribute to the final performance.

In Bert-Transformer(ToBERT), Transformer is the version with location coding, and the importance of location coding

TABLE V
ABLATION EXPERIMENT RESULTS ON NATIVE SPEAKER DETECTION TASK

Models	Accuracy	F1-Score	Precision	Recall
BERT LSTM GAT	83.24%	81.31%	84.90%	80.04%
BERT-base	80.16%	79.15%	79.17%	79.13%
BERT GAT	80.02%	78.84%	79.12%	78.62%

to the accuracy of classification prediction has been proved in previous experiments. The GAT network is for a new attention mechanism, while the LSTM network is to compensate for the lack of location information in GAT. Compared with Bert-Transformer, Bert-LSTM-GAT has achieved better results, with a specific range of performance improvement. Compared to the Base BERT, Bert-LSTM-GAT outperformed because the hierarchical structure makes it possible to obtain more information from the original data. Meanwhile, the hierarchical structure reduces the computational complexity.

What's more, ablation experiments were also performed on multiple datasets. The results obtained from ablation experiments are shown in Table VI. The evaluation index adopted here is accuracy.

TABLE VI
ABLATION EXPERIMENT RESULTS ON OTHER CLASSIFICATION TASKS

Models	IMDB	20 News	AG news	WOS
BERT LSTM GAT	94.11%	84.82%	93.42%	94.21%
BERT-base	89.19%	82.70%	94.18%	88.76%
BERT GAT	92.56%	85.06%	93.20%	94.19%

The performance of all datasets except AG news is improved with the support of the hierarchical structure. Because AG's text length is generally short, basic BERT is adequate for coverage. However, the hierarchical structure will cause some self-attention information loss so that the performance will decrease.

As seen from the results in Table VI, the contribution of the LSTM network, that is, location order information, to the overall structure should not be ignored. Except for the 20NewsGroup dataset, the accuracy of all datasets declined without the LSTM network. As with the above result, the performance of the model with the LSTM network is decreased in the 20NewsGroup dataset.

V. CONCLUSION

Compared with native-speaker voice detection, native speaker writing detection lacks information outside the text, so it is a challenging job to judge only from inside the text. In this work, we proposed a hierarchical structure for native speaker writing detection to solve the problem of the limited length of input text sequence in BERT. Our model divides a long text into shorter segments, uses BERT to build segment vectors, and then extracts the temporal information and the interaction between segment vectors to build document vectors step by step. We have carried out experiments on a self-built native speaker writing detection dataset and obtained good results. We also evaluated various classification tasks, including sentiment analysis, news classification, and topic classification. Moreover, we use these four datasets: IMDB, 20 newsgroups, AG News, and WOS. This method has achieved a good performance in these datasets.

ACKNOWLEDGMENT

This research was supported by the National Nature Science Foundation of China (61872288).

REFERENCES

[1] Y. Shibuya, "Differences between native and non-native speakers' realization of stress-related durational patterns in American English", J.acoust.soc.am, 1996, 100(4):2725-2725.
[2] A. Stephenson, N. Wall, "A Performance Comparison of Native and Non-native Speakers of English on an English Language Proficiency Test."
[3] Y. Gong, Z. Chen, I. -H. Chu, P. Chang and J. Glass, "Transformer-Based Multi-Aspect Multi-Granularity Non-Native English Speaker Pronunciation Assessment," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 7262-7266.
[4] K. Radha, M. Bansal and S. M. Shabber, "Accent Classification of Native and Non-Native Children using Harmonic Pitch," 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), 2022, pp. 1-6.
[5] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding", 2018.
[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). 2017.
[7] N. Chomsky. "Aspects of the Theory of Syntax". The MIT Press, 1965.
[8] H. Peng, J. Li, S. Wang, L. Wang, Q. Gong, R. Yang, "Hierarchical Taxonomy-Aware and Attentional Graph Capsule RCNNs for Large-Scale Multi-Label Text Classification," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 6, pp. 2505-2519, 2021.
[9] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel and N. Dehak, "Hierarchical Transformers for Long Document Classification," 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2019, pp. 838-844.
[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory", Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.
[11] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks" in 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018.
[12] M. E. Ayadi, M. S. Kamel, and F. Karray. "Survey on speech emotion recognition: Features, classification schemes, and databases." Pattern Recognition 44.3(2011):572-587, 2011.
[13] H. Teager, "Some observations on oral air flow during phonation", IEEE Trans. Acoust. Speech Signal Process. 28 (5) (1990) 599–601, 1990.
[14] A. P. Singh, R. Nath and S. Kumar, "A Survey: Speech Recognition Approaches and Techniques," 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2018, pp. 1-4,
[15] C. Sun, A. Myers, C. Vondrick, K. Murphy and C. Schmid, "VideoBERT: A Joint Model for Video and Language Representation Learning," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 7463-7472.
[16] G. Li, N. Duan, Y. Fang, M. Gong, D. Jiang and M. Zhou, "Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training", AAAI, pp. 11336-11344, 2020.
[17] S. Jones, "A statistical interpretation of term specificity and its application in retrieval", Journal of Documentation, 1972