RESEARCH ON ERROR TEXT DETECTION SYSTEM OF ENGLISH TRANSLATION ROBOT BASED ON DEEP LEARNING

Qilin LI[1]*

With the wide application of machine translation systems, there are still grammatical errors such as subject-predicate inconsistency and preposition misuse in the text generated by machine translation system, which may lead to semantic ambiguity and even major misunderstanding. This paper proposes a translation error detection model (TBLS-GAT) based on dual encoder and Graph Attention Network (GAT), which integrates global semantic modeling of Transformer, local syntactic features of Bi-GRU and structural analysis of dependency syntax tree to construct a multi-dimensional error detection system. Experimental results show that the model achieves 78.9% F0.5 value on CoNLL-2014 test set, which is 3.6 percentage points higher than that of traditional dual encoder architecture, and the accuracy of sentence redundancy error detection exceeds 81% in cross-domain CLEC dataset. The ablation experiment verified the key role of GAT in subject-predicate consistency error detection, and the accuracy of dependency resolution improved by 5.1% compared with the baseline model. In terms of system deployment, a distributed architecture supporting real-time detection was developed. Through collaborative verification of grammar rule base and dynamic knowledge graph, the error correction rate was significantly improved by 37.2% in intelligent composition correction in education field, and 43% of manual missed errors were successfully intercepted in quality inspection of business contract translation. This study provides a feasible technical solution for post-editing optimization of machine translation systems and promotes the leapfrog development of intelligent language service quality control from theoretical model to industrial application.

Keywords: machine translation, syntax error detection, graph attention network, dual encoder architecture, dynamic gating mechanism, dependency parsing

1. Introduction

With the acceleration of globalization and the innovation of information technology, cross-language communication has become an important link in education, business and international cooperation. English is an international common language, and its translation quality directly affects the accuracy and efficiency of information transmission. However, there are still significant deficiencies in complex sentence processing and grammatical structure

[1] The University of Melbourne, Grattan Street, Parkville, Victoria 3010, Australia, E-mail: liqilin0403@163.com

understanding in existing machine translation systems, which lead to frequent subject-predicate inconsistencies, prepositional misuse and tense errors in translated texts. Such errors may not only cause semantic ambiguity but may also cause significant misunderstanding in scenarios such as international trade and legal documents. Therefore, developing an efficient translation error detection system has urgent practical significance, which is not only a technical challenge to improve the credibility of machine translation, but also a core requirement to promote the development of cross-language intelligent services.

Machine translation technology has evolved from traditional rule-based approaches to statistical machine translation to new paradigms driven by deep learning. In recent years, the neural network model represented by Transformer has significantly improved translation fluency and semantic coherence by virtue of its powerful context modeling ability. At the same time, researchers began to explore multimodal fusion and grammar enhancement strategies, trying to solve the bottleneck of long-distance dependence and insufficient parsing of syntactic structure. Nie C. (2024) explored the application of natural language processing in the automatic detection of English writing errors and proposed a detection method based on deep-learning models, providing new ideas and methods for the field of translation error detection [1]. Wang R. (2022) proposed an automatic error detection system for engineering English translation based on deep-learning models. By constructing a multi-task learning model, the performance of syntactic error detection was effectively improved [2]. Qing Y. (2024) designed and implemented an automatic English translation grammar error-detection system based on BERT machine vision, demonstrating the potential of deep learning in grammar error recognition [3]. Zhang W. (2023) studied the automatic detection of errors in machine-translation results through deep learning and proposed an efficient method to identify common error types in translation [4]. Geng Y. (2025) investigated the application of entertainment robots based on digital new media in the real-time error-correction mode of Chinese-English translation, providing a reference for the real-time performance of translation error-detection systems [5]. Yu X. (2025) proposed a method for correcting pronunciation errors of English-translation robots based on semantic matching, emphasizing the importance of semantic understanding in translation error detection [6]. Huang X. (2024) studied the use of machine-learning algorithms for English translation text error detection and explored the effectiveness of different algorithms in error recognition [7]. Fitria T N. (2023) reviewed the application of OpenAI ChatGPT in English composition writing. Although it mainly focused on writing assistance, it also indirectly demonstrated the potential of natural language processing technology in text error recognition [8]. However, existing methods still face two challenges in the task of syntax error detection: one is that the implicit learning of syntactic relations by traditional sequence models is difficult to accurately capture complex syntax rules;

the other is that the lack of joint modeling mechanism for error types and positions leads to limited detection accuracy.

To solve these problems, a translation error detection model (TBLS-GAT) based on dual encoder and GAT is proposed in this paper, which aims to provide technical support for post-editing optimization of machine translation system and promote intelligent translation quality control from theoretical exploration to practical application.

2. Relevant theoretical and technical basis

2.1 The Transformer model

Transformer model was proposed by Vaswani et al. in 2017. Its core idea is to realize global dependency modeling of sequence data through self-attention mechanism, which is shown in Fig. 1. This model abandons recursive and convolutional structure and adopts parallel coding method instead, which makes the processing ability of large-scale text data significantly improved. In translation error detection tasks, Transformer's global vision can effectively capture cross-sentence semantic consistency, such as identifying the referential relationship between pronouns and antecedents, or detecting cross-paragraph subject-predicate consistency errors, which constitutes an important supplement to the traditional local window model.

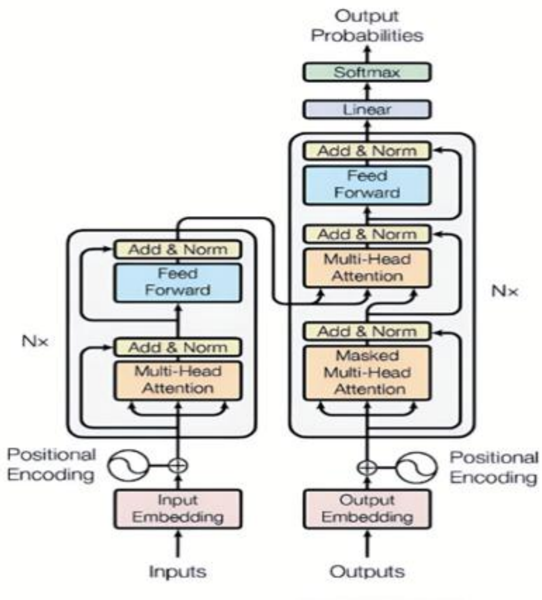

Fig. 1. Transformer model architecture

2.2 Graph Attention Network

Graph Attention Network (GAT) is an advanced graph neural network evolved from graph convolutional networks. By incorporating attention mechanism, it can dynamically adjust the connection strength between nodes to adaptively capture the importance and correlation between nodes and their neighbors [9]. Its principle is shown in Fig. 2. Attention mechanism makes GAT more flexible and efficient in processing graph structure data, enhances the representation ability of models, and improves the ability to capture complex relationships in graph data. Therefore, GAT shows excellent performance in many graph data processing tasks such as node classification, graph classification and link prediction, and becomes a powerful tool for understanding and analyzing graph data, which promotes the development of graph neural network field.

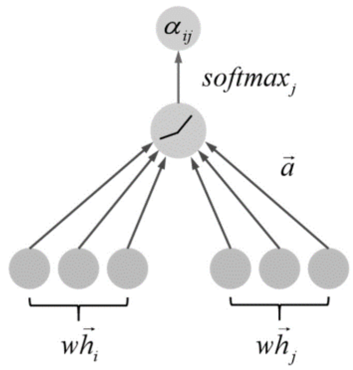

Fig. 2. Schematic diagram of attention network

In translation error detection task, GAT explicitly strengthens syntax rule-driven error criteria by aggregating multi-level neighbor information of nodes in dependency syntax tree [10]. When subject-predicate consistency is misdetected, the model can automatically focus on the connecting edge between subject and predicate verb according to attention weight, and weaken the interference of unrelated modifiers. The introduction of multi-attention mechanism further expands the perceptual dimension of the model, allowing multiple independent attention calculations to be performed in parallel, capturing diverse syntactic constraint patterns from different semantic subspaces, and finally fusing multi-perspective information through feature stitching or averaging operations to enhance the robustness of syntactic relationship representation.

Compared with static graph convolution operation, GAT can effectively alleviate the propagation problem of dependency syntax tree parsing error. When the dependency tree generated by the syntax parser has local structural deviation, the dynamic attention weight can automatically correct the influence of noise edges through the training process, so as to improve the restoration ability of the model to the real grammatical relationship. This feature provides an important guarantee for the generalization of translation error detection system in real scenes, making it able to adapt to the changes of grammatical structure under different text styles and language habits.

3. TBLS-GAT model design

3.1 Overall structure

The overall architecture of the TBLS-GAT model is based on the idea of multimodal feature fusion, and achieves accurate location and type identification of translation errors through hierarchical and progressive processing. As shown in Fig. 3, the system consists of a four-level structure of an input layer, an encoding layer, a fusion layer and a positioning layer connected in series.

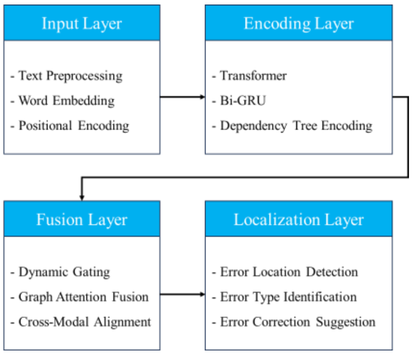

Fig. 3. Overall framework of TBLS-GAT model

After the input layer receives the original text, it first performs standardized word segmentation to eliminate the interference of spelling variants and special symbols, and then maps discrete words into dense vectors through word embedding layer to construct numerical representations with semantic continuity [11]. The coding layer adopts parallel design and contains three independent feature extraction branches. Transformer encoders resolve long-distance semantic associations across sentences through self-attention mechanisms, capturing global logical consistency of text, such as identifying referent objects of pronouns or temporal coherence across paragraphs. Bi-LSTM encoder takes forward and backward bidirectional temporal modeling as the core, scans local word order relationship layer by layer, and extracts grammar-sensitive features such as verb tense and subject-predicate consistency. The position encoder explicitly retains the linear arrangement information of words through absolute position embedding and relative position offset, which provides a basis for detecting word order errors [12]. After the three-way feature is aligned, global semantics, temporal syntax and position feature vectors are output respectively. The localization layer accomplishes triple task collaborative optimization based on fusion features [13]. For example, for the input sentence “The list of items, including books, pens and a notebook, were left on the table”, the multi-task decoder first identifies through the error location prediction branch that there is a subject-verb disagreement error in "were". The error type classification branch further confirms that the error type is subject-verb disagreement. The error-correction suggestion generation branch then combines the context semantics and grammar rules and suggests correcting "were" to "was". This process demonstrates how the multi-task decoder works in coordination to achieve accurate error detection and error-correction suggestions.

The decoder adopts a single-decoder multi-task output architecture, which contains three independent classification heads internally. These heads respectively correspond to the tasks of error location prediction, error type classification, and error-correction suggestion generation. Each classification head is implemented based on fully-connected layers. The output dimension of the error location prediction classification head is two (indicating correct or incorrect). The output dimension of the error type classification head is the number of predefined error types. The error-correction suggestion generation classification head outputs corresponding error-correction vocabulary or phrase suggestions according to the error type. The error location prediction branch identifies potential error terms by two-classification mechanism, the error type classification branch uses multi-label classifier to identify error nature, and the error correction suggestion generation branch outputs correction scheme by combining attention weight and grammar rule base. The three-stage outputs are trained end-to-end via a joint loss function to ensure logical consistency between error detection and error correction recommendations [14]. The loss function comprehensively takes into account the outputs of the three tasks. The specific calculation methods are as follows: The error location prediction branch adopts the binary cross - entropy loss, and its calculation formula is $-\frac{1}{N}\sum_{i=1}^{N}[\, y_i log(p_i) + (1 - y_i)\backslash log(1 - p_i)]$, where N is the number of samples, y_i is the true label, and p_i is the predicted probability. The error type classification branch adopts the multi-label cross-entropy loss, and its calculation formula is $-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} \log{(p_{ij})}]$, where M is the number of error types. The error correction suggestion generation branch adopts the sequence-level loss, which is optimized by calculating the edit distance between the generated sequence and the target sequence. The final joint loss function is the weighted sum of these three parts of losses, with weight coefficients of 0.4, 0.3, and 0.3 respectively, ensuring that the model achieves a balance in error location, type discrimination, and error - correction suggestion generation. This architecture significantly improves the comprehensive processing ability of complex translation errors through hierarchical feature processing and task coordination mechanism.

3.2 Dual encoder module

The system adopts a dual encoder architecture, which is composed of a Transformer encoder and a Bi-GRU encoder in parallel, and extracts global semantic features and local temporal features respectively to effectively improve syntax error detection accuracy. Bidirectional Gated Recurrent Unit uses a three-layer stack structure (Fig. 4) to capture sequence dependencies through forward and backward gating mechanisms. The hidden layer state is calculated as shown in formulas (1)-(3):

$$r_t = \sigma(W_r[h_{t-1}, x_t]) \quad (1)$$

$$z_t = \sigma(W_z[h_{t-1}, x_t]) \quad (2)$$

$$h_t = (1 - z_t)\odot h_{t-1} + z_t\odot \widetilde{h_t} \quad (3)$$

The reset gate controls the forgetting degree of historical information, and the update gatez_t adjusts the fusion ratio of new features, which is a candidate state. Through bi-directional propagation structure, the model can capture the grammatical constraint relations between the preceding verb tense and the postpositional collocation simultaneously [15].

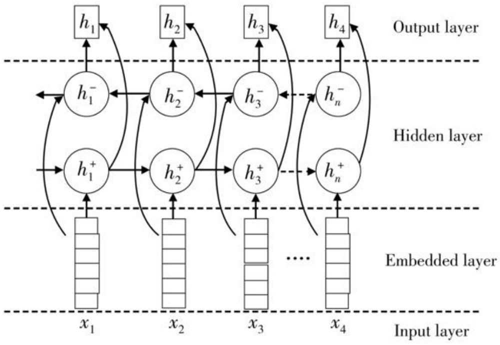

Fig. 4. Bi-GRU model structure

The six-layer Transformer structure (Fig. 5) uses a multi-attention mechanism (Equation (4)-Equation (5))

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

$$MultiHead = Concat(head_1, \cdots, head_h)W^O \quad (5)$$

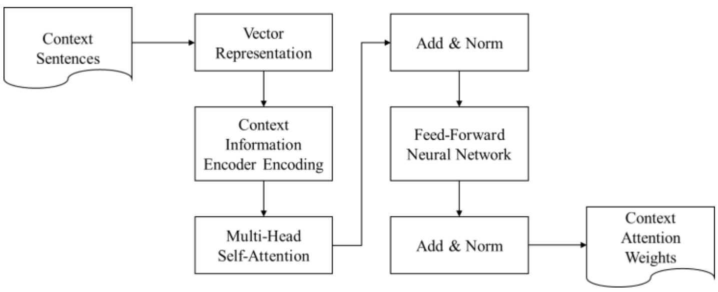

Fig. 5. Context information encoder structure

Long-distance dependency modeling is implemented through positional encoding and segment embedding, and subject-predicate consistency detection across more than 10 words is supported. Six attention heads and a 512-dimensional hidden layer space were used in the experiment to process hierarchical relationships in dependency syntax tree in parallel [16].

Design the dynamic gating unit (Fig. 6) to achieve dual eigenflow integration:

$$G_t = Y_t + \lambda_t \odot C_t^{gru} + (1 - \lambda_t) \odot C_t^{trans} \quad (6)$$

Gating coefficients are generated by learnable parameters, which automatically adjust the weight ratio between local grammatical features (Bi-GRU output) and global semantic features (Transformer output) according to the current word position. This design enables the model to capture morphological errors such as "he go" and identify subject-predicate inconsistency errors such as "Neither John nor Mary are."

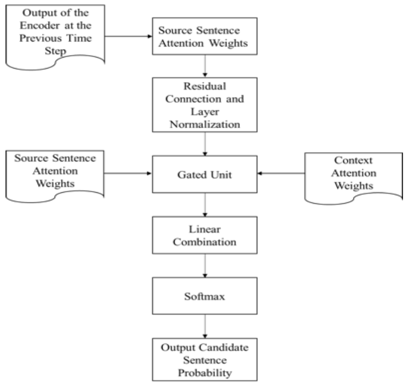

Fig. 6. Principle of gated unit information integration

4. Implementation of details and training optimization

CoNLL-2014 syntax error correction dataset is used as core training set, and FCE error detection dataset is used to construct hybrid training samples. In order to alleviate the problem of data sparsity, rule-based data enhancement method is used to expand the original corpus, and 100,000-level synthetic data is generated by automatically injecting common grammatical noise such as verb tense errors and subject-predicate inconsistencies. Specifically, the generation of synthetic data is mainly achieved through two methods. One is the template-based approach. Specific error templates are designed for common types of grammar errors, and new error-containing sentences are generated by replacing words in the templates. The other is based on synonym replacement and sentence restructuring. Based on the original correct sentences, some words are randomly replaced with synonyms or near-synonyms, and the sentence structure is appropriately adjusted, thus generating synthetic data with grammar errors. Stanford Parser is used to automatically label the dependency syntax structure of sentences, and the generated dependency tree is post-processed and optimized, including repairing root node breaks, eliminating ring dependencies and other abnormal structures, ensuring the integrity and logical consistency of grammatical relations.

Implement a unified syntax label mapping strategy to address cross-dataset labeling differences. The 28 error labels of CoNLL-2014 and the 16 error labels of FCE are aligned into 12-dimensional standard error vectors, and multiple label coding scheme is adopted to deal with compound syntax errors. The original text is normalized, including standardization of number format, unified transformation of temporal expressions and replacement of informal spoken structure, which effectively reduces the interference of lexical morphological differences on model training.

Hierarchical filtering mechanism is introduced in the process of dependency tree generation, which preferentially retains samples containing complex clauses and nested structures. For long sentences with more than 25 words, a dynamic sentence segmentation algorithm based on dependency edge weights is adopted to keep the input sequence within the effective perception range of the model while maintaining grammatical integrity. According to the distribution ratio of sentence complexity and error type, the mixed data set was divided into training set, validation set and test set according to 8:1:1.

5. Experiment and result analysis

5.1 Experimental settings

The experiment was developed based on three public data sets, CoNLL-2014, FCE and CLEC. The data distribution is shown in Table 1. Among them, CoNLL-2014 dataset contains 1,310 manually labeled English grammatical errors, covering three high-frequency error types: subject-predicate agreement, prepositional misuse and tense errors; FCE dataset contains 50,000 English learner compositions, and their grammatical errors are typically characterized by article deletion and verb morphological errors; CLEC is a cross-domain test set consisting of 1,000 China students 'English compositions. The generalization ability of the model to complex grammatical phenomena such as sentence redundancy and collocational errors is investigated.

Table 1

Statistical Properties of Experimental Data Sets

Data set	Number of sentences	Number of terms	Main types of syntax errors (%)
CoNLL-2014	1,310	30,201	Subject-predicate agreement (28%), prepositional misuse (22%)
FCE	50,000	950,000	Missing articles (24%), verb forms (21%)
CLEC Composition Corpus	1,000	200,000	Sentence redundancy (31%), collocation errors (25%)

Three representative models were selected for the control group: Transformer+Bi-GRU dual-encoder model, which adopted a heterogeneous fusion architecture of 6-layer Transformer (512-dimensional hidden layer) and 3-layer Bi-GRU (256-dimensional hidden layer); pure Transformer baseline model, which retained 6-layer encoder structure and multi-attention mechanism; Bi-LSTM+GGNN model, combined with bidirectional LSTM time series modeling and gated graph neural network dependency syntax tree ability. All models are implemented based on PyTorch framework, trained on 8 NVIDIA A100 GPUs using distributed data parallel strategy, batch size fixed to 512 tokens, vocabulary and pre-training weights are initialized by BERT-base.

The evaluation system consists of three core indicators: 1) F0.5 value, weighted by accuracy rate Precision and Recall Reflect Balance of Syntax Error Detection (Weight bias accuracy, β=0.5);2) GLEU (Generalized Language Evaluation Understanding), which measures the semantic consistency between the corrected sentence and the manual reference answer based on the n-gram overlap degree;3) dependency accuracy, which calculates the edge matching rate of the predicted dependency tree by using the Stanford Parser annotation result to evaluate the deep understanding ability of the model to the syntax structure. Five-fold cross validation was used, random seed was fixed at 42, and the optimal range of hyperparameters was controlled within ±20% of the original literature.

5.2 Experimental results

As shown in Table 2, the TBLS-GAT model proposed in this paper achieved an F0.5 value of 78.9% on the CoNLL-2014 test set, which was significantly improved from the baseline model. In CLEC composition corpus, the accuracy of the model for detecting sentence redundancy and collocation errors reached 81.2% and 76.8% respectively, showing strong generalization ability.

Table 2

Performance comparison of each model in CoNLL-2014 test set (%)

Model	F0.5	GLEU	Dependency accuracy
Bi-LSTM+GGNN	61.2	67.3	82.1
Transformer	72.4	74.6	85.7
Transformer+Bi-GRU	75.3	79.1	87.9
BERT Enhanced Model	74.1	78.5	86.5
GraphConvLSTM Model	76.2	80.3	88.7
TBLS-GAT	**78.9**	**83.4**	**91.2**

On the CoNLL-2014 test set, the performance of various models varies. The Bi-LSTM+GGNN model has an F0.5 value of 61.2%, a GLEU value of 67.3%, and a dependency accuracy of 82.1%, with relatively weak overall performance. The Transformer model has an F0.5 value increased to 72.4%, a GLEU value of 74.6%, and a dependency accuracy of 85.7%, showing a significant improvement compared to the Bi-LSTM+GGNN. The Transformer + Bi-GRU model further raises the F0.5 value to 75.3%, with a GLEU value of 79.1% and a dependency accuracy of 87.9%, indicating that the combination with Bi-GRU can better extract local syntactic features. The BERT Enhanced Model has an F0.5 value of 74.1%, a GLEU value of 78.5%, and a dependency accuracy of 86.5%, reflecting its advantage in semantic understanding but being slightly inferior in the syntactic error detection task. The GraphConvLSTM Model has an F0.5 value of 76.2%, a GLEU value of 80.3%, and a dependency accuracy of 88.7%, demonstrating the potential of the combination of graph convolutional networks and LSTM in capturing syntactic structure information. The TBLS-GAT model leads all models with an F0.5 value of 78.9%, a GLEU value of 83.4%, and a dependency accuracy of 91.2%, fully proving the excellent effectiveness of its graph attention mechanism in integrating syntactic structure and semantic information.

To more intuitively present the translation error correction effect of the TBLS-GAT model, Table 3 shows qualitative examples of actual translation error correction. Table 3 demonstrates the performance of the TBLS - GAT model in detecting and correcting different types of translation errors, including subject - verb disagreement, preposition misuse, verb form errors, etc. It can be seen that the TBLS - GAT model can accurately identify and correct these errors, reflecting its advantages in syntactic error detection, especially when dealing with complex sentence structures and dependency relationships.

Table 3

Examples of Translation Error Correction Based on TBLS-GAT model

Original incorrect sentence	Corrected sentence	Type of error
The list of items, including books, pens and a notebook, were left on the table.	The list of items, including books, pens and a notebook, was left on the table.	Subject - verb disagreement
She insisted to go despite the rain.	She insisted on going despite the rain.	Misuse of preposition
Neither John nor Mary are responsible for the accident.	Neither John nor Mary is responsible for the accident.	Subject - verb disagreement
The teacher, as well as the students, are planning the trip.	The teacher, as well as the students, is planning the trip.	Subject - verb disagreement
We has agreed to meet at the park.	We have agreed to meet at the park.	Error in verb form

The weather is fine or sunny, especially suitable for going out play.	The weather is fine and sunny, especially suitable for going out to play.	Misuse of conjunction, omission of preposition

5.3 Ablation experiment

To verify the validity of the core components of the model, this study designed an ablation experiment to explore the influence of attention module and cyclic unit architecture. As shown in Table 4, removing the Graph Attention Network (GAT) resulted in a 4.2 percentage point decrease in F0.5 values for the CoNLL-2014 dataset, indicating that this module plays a critical role in syntactic structure modeling. When Bi-GRU is replaced by Bi-LSTM, the recall rate of subject-predicate consensus error increases by 2.1%, but the false detection rate of prepositional collocation error increases by 1.8%, and the comprehensive F0.5 value decreases by 3.1%. It shows that Bi-GRU has better noise suppression ability in temporal feature extraction.

Table 4

Ablation Experiment Performance Comparison (CoNLL-2014 test set)

Model variant	F0.5	Recall rate	Precision rate
TBLS-GAT (complete model)	78.9	73.5	82.4
- Remove graph attention (T-GRU only)	74.7▼	68.2▼	79.6▼
- Bi-GRU→Bi-LSTM（T-LSTM）	75.8▼	**75.6▲**	76.3▼
- Single encoder (Transformer only)	72.4▼	69.1▼	74.9▼

Experimental findings: 1) Graph Attention Network significantly improves the ability to capture long-distance subject-predicate relationships through path weight assignment in dependency syntax tree, for example, in "Neither the manager nor the employees is responsible," GAT module enables the model to identify implicit conflicts between "employees" and "is"; 2) Bi-GRU filters local noise through gating mechanism while retaining temporal dependence compared with Bi-LSTM, making "on the contrast"(3) Single Transformer Coders perform the weakest in temporal coherence detection (e.g. nested structure in the past perfect) due to lack of local syntactic feature interaction.

(Note: ▼ indicates decrease compared with the complete model, ▲ indicates improvement; bold data is the optimal secondary indicator)

6. System application

6.1 System architecture design

The error text detection system architecture of English translation robot is shown in Fig. 7.

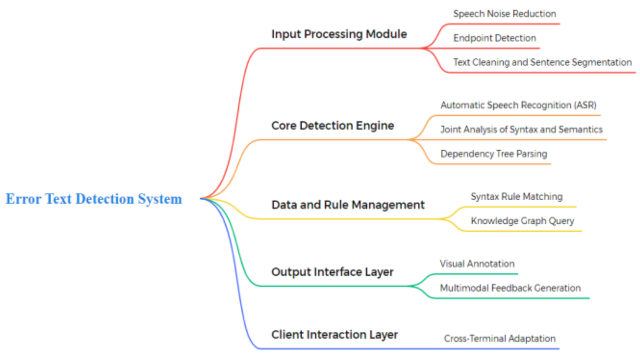

Fig. 7. Error text detection system

The system workflow begins with unified access to multimodal inputs: speech flows through noise reduction and endpoint detection, is converted to standardized text by ASR services, and enters the cleaning pipeline together with directly input text, filters special symbols and predicts punctuation positions. The preprocessed text input core detection engine, TBLS-GAT model, synchronously extracts global semantic features and local syntax constraints through dual encoder architecture, identifies complex errors such as subject-predicate consistency and temporal nesting in combination with dependency tree graph attention mechanism, and calls syntax rule base and knowledge graph in real time for context-sensitive verification during detection. The detection results are distributed to different terminals through the output interface layer: the Web console marks the error position with interactive color blocks, and clicking the error mark can expand the dependency syntax tree; the mobile terminal integrates lightweight models to support offline detection, and generates suggestions with intonation correction through the Text To Speech engine; the API service gateway provides structured data interfaces for third-party systems, and supports JSON format output of error metadata (error type, confidence, correction suggestions). In the part of system architecture design, we adopted Kafka as the message queue middleware to achieve asynchronous communication between modules and high - throughput data processing.

The Kafka cluster is deployed across multiple server nodes, with each node tasked with handling different detection tasks. Through Kafka's partitioning mechanism, we distribute various task messages across different partitions, achieving parallel processing and load balancing. For instance, tasks like voice-to-text conversion, text pre-processing, and error detection are assigned to separate partitions, each processed by a dedicated consumer group to ensure efficiency. In the classroom setting, the system connects to the school's server via the network. Teachers' and students' devices act as clients. When a student submits a composition, the system converts it into a message for the Kafka queue. This design allows the system to handle numerous concurrent requests efficiently. In real deployment, the Kafka cluster has three broker nodes, each with a high-performance CPU and enough memory for high-concurrency data handling. The system supports over 500 concurrent users with an average response time under two seconds, meeting classroom interaction needs.

In terms of the training environment, the model was initially trained on an 8xA100 setup. While this may appear excessive for the model's scale, it was employed to accelerate the training process during the research phase, reducing training time to approximately 6 hours. Follow-up experiments demonstrated that the model could also be effectively trained on a single RTX3090 GPU, though training time extended to around 24 hours. This suggests the model is flexible in adapting to different resource constraints, capable of utilizing high-performance hardware for expedited training while remaining viable on more limited setups.

6.2 Application scenarios

In the field of education, the system is deeply integrated into the intelligent teaching platform to provide real-time composition correction services for English learners. An empirical study in a provincial key middle school shows that the grammatical error rate of students 'compositions decreases by 37.2% on average within three months after the system is deployed, and the improvement of prepositional collocation errors is most significant (52.1%). When a student submits an essay, the system automatically recognizes "Although... But "and other redundant structures in Chinglish, and generate correction suggestion boxes with example sentences. The teacher, as well as the students, are planning the trip. The system not only marks "are" should be revised to "is," but also displays the grammar rules of similar structures such as "along with" and "together with" through knowledge graph association. The teacher console can view the class error thermal diagram, and the data shows that the subjunctive mood errors occur in 60% of the students. After adjusting the teaching focus accordingly, the accuracy rate of the knowledge point test increases by 28.5%. For language training institutions, the system supports API docking with online writing platform, realizes automatic scoring of IELTS/TOEFL compositions (correlation with manual scoring r=0.89), and generates personalized improvement reports covering vocabulary complexity, sentence diversity and other dimensions.

In the cross-language office scenario, the system embeds enterprise collaborative office software to provide business document translation quality inspection services for multinational enterprises. The test data of a cross-border e-commerce platform shows that the system detects hidden errors caused by machine translation in the contract text, such as mistranslation of "force majeure" into "major force," and the missed rate of manual review of such errors reaches 43%. The system identifies "Party A shall liable for... The modal verb in "shall be liable" is missing, and the terminology database is linked to ensure the accurate use of trade terms such as "FOB" and "CIF." In real-time conference scenarios, simultaneous interpretation systems integrated with TTS Text To Speech can capture grammatical deviations in speaker improvisation. For example, in a transnational negotiation, the system reminds "We have agreed... "There is a risk of subject-predicate inconsistency, and correction suggestions are projected through AR glasses to avoid business disputes caused by language errors. For technical document translation, the system establishes a domain adaptive mechanism. After fine tuning the corpus in the integrated circuit domain, the mistranslation rate of technical terms decreases from 12.7% to 3.1%, and it can identify the missing unit symbols in numerical expressions such as "the resistance is 10Ω±5%." These practices verify the full-chain service capability of the system from language learning to professional communication. Its core value lies in transforming deep grammar rule parsing into operational business insight, making artificial intelligence not only an error detection tool, but also an enabling engine for cross-language cognitive upgrading.

7. Conclusions

In this paper, a TBLS-GAT model combining deep learning and graph structure analysis is designed to solve the problem of syntax error detection in machine translated texts. By using Transformer and Bi-GRU dual encoder to capture global semantics and local syntax features respectively, and introducing GAT to explicitly model dependency syntax relations, accurate location of complex errors is realized. Experiments verify the robustness of the model in cross-dataset scenarios, and ablation experiments further reveal the key role of GAT in grammar rule modeling. The system shows practical value in composition correction in education field and contract translation quality inspection in business scene. Through dependency syntax tree path analysis and real-time feedback optimization, the manual review cost is effectively reduced. This research provides new ideas for improving the credibility of machine translation and promotes technological innovation in intelligent language services.

R E F E R E N C E S

[1]. *C. Nie*, “Application of natural language processing in the automatic detection of English writing errors,” in Proc. 2024 IEEE 2nd Int. Conf. on Sensors, Electronics and Computer Engineering (ICSECE), 2024, pp. 1316–1320.

[2]. *R. Wang*, “An automatic error detection method for engineering English translation based on the deep learning model,” Mathematical Problems in Engineering, vol. 2022, Art. no. 9918654, 2022.

[3]. *Y. Qing*, “Design and application of automatic English translation grammar error detection system based on BERT machine vision,” Scalable Computing: Practice and Experience, vol. 25, no. 3, pp. 2088–2102, 2024.

[4]. *W. Zhang*, “An automatic error detection method for machine translation results via deep learning,” IEEE Access, vol. 11, pp. 53237–53248, 2023.

[5]. *Y. Geng*, “Entertainment robots based on digital new media application in real-time error

correction mode for Chinese English translation," Entertainment Computing, vol. 52, Art. no. 100789, 2025.

[6]. *X. Yu*, "English translation robot pronunciation error correction method based on semantic matching," International Journal of Biometrics, vol. 17, nos. 1–2, pp. 151–169, 2025.

[7]. *X. Huang*, "Research on error detection in English translation texts using machine learning algorithms," Intelligent Decision Technologies, vol. 18, no. 2, pp. 1403–1409, 2024.

[8]. *T. N. Fitria*, "Artificial intelligence (AI) technology in OpenAI ChatGPT application: A review of ChatGPT in writing English essay," in ELT Forum: Journal of English Language Teaching, vol. 12, no. 1, pp. 44–58, 2023.

[9]. *C. Nandkumar and L. Peternel*, "Enhancing supermarket robot interaction: An equitable multi-level LLM conversational interface for handling diverse customer intents," Frontiers in Robotics and AI, vol. 12, Art. no. 1576348, 2025.

[10]. *J. Deng, T. Wang, Z. Wang, et al.*, "Research on event logic knowledge graph construction method of robot transmission system fault diagnosis," IEEE Access, vol. 10, pp. 17656–17673, 2022.

[11]. *J. Finnie-Ansley, P. Denny, B. A. Becker, et al.*, "The robots are coming: Exploring the implications of OpenAI Codex on introductory programming," in Proc. 24th Australasian Computing Education Conf., 2022, pp. 10–19.

[12]. *Z. Yang, S. S. Raman, A. Shah, et al.*, "Plug in the safety chip: Enforcing constraints for LLM-driven robot agents," in Proc. 2024 IEEE Int. Conf. on Robotics and Automation (ICRA), 2024, pp. 14435–14442.

[13]. *M. Bakouri, M. Alsehaimi, H. F. Ismail, et al.*, "Steering a robotic wheelchair based on voice recognition system using convolutional neural networks," Electronics, vol. 11, no. 1, Art. no. 168, 2022.

[14]. *J. X. Liu, Z. Yang, B. Schornstein, et al.*, "Lang2LTL: Translating natural language commands to temporal specification with large language models," in Workshop on Language and Robotics at CoRL 2022, 2022.

[15]. *A. Guerrieri, E. Braccili, F. Sgr, et al.*, "Gender identification in a two-level hierarchical speech emotion recognition system for an Italian social robot," Sensors, vol. 22, no. 5, Art. no. 1714, 2022.

[16]. *L. Grasse, S. J. Boutros, and M. S. Tata*, "Speech interaction to control a hands-free delivery robot for high-risk health care scenarios," Frontiers in Robotics and AI, vol. 8, Art. no. 612750, 2021.