International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-18

https://doi.org/10.70917/ijcisim-2026-0120
Article

A Study on the Application of Computer-Assisted Sentiment Analysis Methods of Literary Works in Literary and Cultural Education

Ye Wang *

College of Literature (Su Dongpo College), Huanggang Normal University, Huanggang, Hubei, 438000, China; angelahappy2024@126.com

Abstract: This paper addresses the need for emotional analysis of literary works by constructing a customized Chinese novel dataset and proposing a progressive learning framework called PL-ERC. This framework generates pseudo-sentence labels through dialogue-level annotation, combining noise density segmentation strategies with LSTM temporal modeling to achieve fine-grained emotional transfer learning. Cross-domain experiments show that the average F1 score reaches 88.87% in a 10-shot task across seven categories of literary works, outperforming the best baseline ConVEx by 2.9%. Ablation experiments reveal that removing the self-training strategy results in the largest F1 drop of 17.72%, while replacing LSTM with RNN reduces F1 by 12.88%. In terms of sentiment classification accuracy, the accuracy rate on the DMSC seven-classification dataset was 97.17%, with an identification rate of over 98% for the anger/happiness categories. Educational evidence shows that the weighted total score of the experimental class applying this method was 94.25 ± 3.92, significantly better than the control class (84.00 ± 7.89, $p < 0.01$). Students' critical thinking ability scores were 4.67 (control class: 3.51), and aesthetic experience ability scores were 4.72 (control class: 3.74), confirming that technology-enabled education significantly enhances the effectiveness of literary and cultural education.

Keywords: dialogue sentiment analysis; PL-ERC; LSTM; cross-domain transfer; literary and cultural education

1. Introduction

Emotion computing, which involves the use of various computing devices and algorithms to automatically identify, understand, and calculate features related to human emotions, has become a hot topic in cognitive science [1-2]. By utilizing data from multiple modalities such as images, audio, video, and text, computers can perform multi-feature fusion of emotions to recognize human emotions with higher accuracy and achieve high-quality human-computer interaction [3-5]. Emotion computing based solely on text has also found widespread application in fields such as social media and e-commerce platforms [6]. Analyzing the vast amounts of text data generated by these platforms can help governments and businesses quickly and conveniently track public opinion trends or emotional changes regarding a particular event or product, yielding significant social benefits and economic gains, and thus has attracted extensive attention from researchers [7-10].

Literary sentiment analysis provides a quantitative analytical tool, enabling researchers to explore the emotional patterns embedded in literary works in an unprecedented manner [11-12]. This not only helps extend literary computing from word frequency and stylistic analysis to the emotional narrative level but also aids in revealing and summarizing the complex emotional structures within narrative texts, helping us understand how writers construct emotional experiences through language [13-15]. Therefore, from the perspective of cultural communication, the research results of literary emotion analysis can assist in the high-quality development of literary and cultural education.

In recent years, some scholars have attempted to introduce emotion computing methods into the field of literature, conducting quantitative emotion analysis on various literary works, thereby opening up a new field of quantitative literary research. Literature [16] employs topic modeling methods to monitor and visualize the main subjects of literary text corpora, while combining text emotional analysis models to further explore the emotional polarity in the literary works authored by the writer. Literature [17] indicates that literary work evaluation strategies based on emotional analysis technology and social network analysis methods can help clarify the relationships between characters and events in the text, and such detail-oriented analyses play an important auxiliary role in plot adaptation. Literature [18] argues that characters and events are the origins of character emotions. By constructing an emotion analysis model that considers character relationships and plot development at the text level, it predicts character emotional expressions beyond the traditional analysis methods that merely categorize the overall emotional tone of the text, providing valuable insights for future emotion entity recognition. Literature [19] emphasizes the importance of sentiment analysis methods that track plot development and narrative in literary evaluation, while proposing dictionary-based and transformer-based sentiment analysis models to explore solutions to sentiment analysis problems in complex texts. Therefore, the emergence of sentiment computing can be said to have filled a key gap in quantitative literary research and made it possible to construct a higher-level computational criticism theory transcending traditional linguistic models.

This paper constructs a high-quality Chinese novel dataset suitable for literary analysis and proposes an innovative sentiment analysis method that effectively utilizes relatively readily available dialogue-level sentiment annotations to train high-performance sentence-level sentiment analysis models for in-depth semantic analysis of literary texts. This study crawled a large amount of text and its metadata from multiple novel websites. To ensure the representativeness and applicability of the dataset, strict double screening criteria were also established. Through a series of screening processes focusing on the diversity and quality of novels, the appropriateness of dialogue content, and the proportion of pure dialogue, a customized Chinese novel dataset of high quality, diverse types, and rich dialogue content was ultimately obtained. To address the challenge of performing sentence-level sentiment analysis with only dialogue-level sentiment annotations, the paper proposes the innovative PL-ERC framework. The core idea of this framework is to use progressive learning and pseudo-labeling technology to learn fine-grained sentence-level sentiment knowledge from coarse-grained dialogue-level annotations. PL-ERC mainly consists of two key processes. The first is the generation of a training subset, which uses dialogue-level sentiment labels to initialize pseudo-sentence-level sentiment labels for each sentence in the dialogue, forming a pseudo-labeled dataset. The second is gradual updating, which uses a self-training strategy to jointly update the sentiment prediction model and pseudo-labels. Its innovation lies in dividing the data into different subsets based on the “noise density” of the dialogue and progressively training the model in order of increasing noise density. The strong temporal modeling capabilities demonstrated by LSTM in sentiment analysis tasks make it an ideal choice for constructing a predictor that can understand the dynamic changes and context dependencies of sentiment in dialogue flows, providing critical technical support for the effective implementation of the PL-ERC method.

2. A Dialogue Sentiment Analysis Method Based on Progressive Learning PL-ERC

2.1. Selection Criteria for Novels in Literary Works

This paper has selected and created a representative novel dataset for testing and fine-tuning various task models. The novel text data was crawled from multiple novel websites, and some metadata for each novel was also stored, such as category, author, word count, number of clicks, and rating. The number of novels is extremely large and diverse, so it is necessary to perform some filtering to construct an effective and representative dataset. The main screening criteria are the diversity and quality of the novels. Regarding diversity, this paper selects thirteen different types of novels, such as the currently popular fantasy and martial arts novels. Regarding quality, this paper primarily evaluates the quality of a novel by extracting its click-through rate and rating on reading platforms.

Based on the above two screening criteria, this paper has obtained an initial collection of novels that covers various categories and has relatively high quality. Second, since the subsequent in-depth semantic analysis of the novel texts in this paper relies on high-quality dialogue content, this paper conducted a screening based on the proportion of dialogue and pure dialogue in the novels. Dialogue refers to sentences containing dialogue along with non-dialogue narration. Pure dialogue refers to sentences containing only dialogue without narration, where readers must infer the subject of the speech on their own.

The specific screening logic is as follows:

(1) Remove novels with an overly low proportion of dialogue content to ensure sufficient dialogue volume. Many subsequent tasks in this paper require dialogue as data, so ensuring sufficient dialogue volume is essential for the smooth progression of subsequent work.

(2) Remove novels with an overly high proportion of pure dialogue. If the proportion of pure dialogue is too high, the novel becomes difficult for machines to understand. Similarly, an overly high proportion of pure dialogue also poses challenges for human readers.

Based on the above secondary screening, a high-quality novel dataset suitable for the research content of this paper was ultimately obtained.

2.2. Sentiment Analysis Method Based on Dialogue-Level Annotated Learning Models

Dialogue-level annotation schemes are merely coarse-grained annotation methods, capable only of assigning sentiment labels to the entire dialogue, not to each individual sentence within it. However, sentence-level sentiment labels are essential for training actual sentiment classification models. Therefore, traditional dialogue sentiment analysis methods cannot be directly applied to learning dialogue-level annotation schemes. To effectively utilize dialogue-level annotation datasets, this section introduces a dialogue sentiment analysis framework based on progressive learning, named PL-ERC.

2.2.1. Task Definition

To more clearly illustrate the process of training an emotion classification model using dialogue-level annotated samples, this section first provides a task definition. The training dataset consists of N labeled samples $\mathrm{D}=\left\{(C_i,y_i)\right\}_{i=1}^{i=N}$, where C_i represents a conversation, $y_i\in\{0,1\}^{|\mathrm{Y}|}$ is its corresponding sentiment annotation set, and Y represents the sentiment space. Specifically, each C_i consists of M rounds of dialogue $\left\{(u_{ij},s_{ij})\right\}_{j=1}^{j=M}$, where, for simplicity, we assume that all dialogues contain M sentences. Each u_{ij} is the content spoken by the interlocutor s_{ij}. The objective of this task is to train a dialogue sentiment analysis predictor $\mathrm{F}_\theta(\cdot)$ using a training dataset D that contains only the hierarchical sentiment annotation set of dialogues. This predictor is capable of determining the sentiment polarity of individual sentences in subsequent dialogues.

2.2.2. PL-ERC

In order to learn sentence-level sentiment knowledge from labels that only contain overall conversation annotations, this work proposes the PL-ERC framework based on the pseudo-label method. This framework mainly consists of two processes:

(1) Generating the training subset: For each sentence (u_{ij},s_{ij}), we first initialize its pseudo sentence-level sentiment $\hat{y}_{ij}$ using the sentiment set y_i of the dialogue to which it belongs, as shown in Formula (1):

$$\hat{y}_{ij}=\frac{y_i}{|y_i|},\quad i=[N],j=[M] \tag{1}$$

where $|y_i|$ denotes the number of emotions assigned to y_i. Once all sentences in the dialogues have been assigned pseudo labels, this method can initialize a pseudo dialogue sentiment analysis dataset $\widehat{\mathrm{D}}=\left\{\left(C_i,\left\{\hat{y}_{ij}\right\}_{j=1}^{j=M}\right)\right\}_{i=1}^{i=N}$ that is identical in format to normal dialogue-level sentiment analysis.

(2) Iterative update: Based on $\widehat{\mathrm{D}}$, this method jointly updates $\hat{y}$ and a dialogue sentiment analysis predictor $\mathrm{F}_\theta(\cdot)$ with parameter θ in a self-training manner. After each model update, the current dialogue sentiment analysis predictor's prediction $p_{ij}=\mathrm{F}_\theta(u_{ij},s_{ij})$ is used to update the noisy pseudo labels $\hat{y}_{ij}$. To achieve more accurate $\hat{\mathbf{y}}$ during early training, we divide $\widehat{\mathrm{D}}$ into several training subsets with different noise densities, and then gradually update the model from the training subset with lower

noise density to the training subset with higher noise density using $\mathrm{F}_{\theta}(\cdot)$. The following sections will provide a more detailed description of the process of generating training subsets and gradually updating the model.

2.2.3. Generating the Training Subset

Since each sentence's pseudo-label sentiment $\hat{y}_{ij}$ must be covered by its corresponding dialogue-level sentiment set y_i, this work can use the number of sentiments in the dialogue-level sentiment set $|y_i|$ to represent the overall noise density from the current dialogue C_i. and divide $\widehat{\mathrm{D}}$ into several disjoint training subsets $\widehat{\mathrm{D}} = \widehat{\mathrm{D}}_1 \cup \cdots \cup \widehat{\mathrm{D}}_{|\mathrm{Y}|}$ based on the size of the noise density. It should be noted that dividing in this way may result in empty sets, because for real data, there may be no conversations with certain specific numbers of emotions. Subsequently, for each subset $\widehat{\mathrm{D}}_g$, define as shown in Formula (2):

$$\widehat{\mathrm{D}}_g = \left\{ \left(C_i, \left\{ \hat{y}_{ij} \right\}_{j=1}^{j=M} \right) \right\}_{i=1}^{i=N_g}, \forall |y_i| = g \tag{2}$$

where N_g is the number of sentences in the current subset. Based on this definition, this work can obtain the noise density of the entire conversation, which also reflects the difficulty of learning sentences in the conversation to a certain extent, because the more emotions there are in the conversation, the more difficult it is to match these emotions with the sentences in the conversation one-to-one.

2.2.4. Gradual Updating

Given the training subset, PL-ERC starts with dialogues with lower noise density and gradually learns to handle dialogues with higher noise density, i.e., it updates the dialogue sentiment analysis predictor $\mathrm{F}_{\theta}(\cdot)$ from $\widehat{\mathrm{D}}_1$ to $\widehat{\mathrm{D}}_{|\mathrm{Y}|}$. Meanwhile, PL-ERC updates the noisy sentiment pseudo-labels $\hat{y}$ for sentences in the dialogue. Specifically, the model performs the following learning process:

(1) Initialize a training pool $\widetilde{\mathrm{D}}$ using $\widehat{\mathrm{D}}_1$, and jointly update $\hat{y}$ and $\mathrm{F}_{\theta}(\cdot)$ using $\widetilde{\mathrm{D}}$ over T cycles.

(2) For $t = 2, \cdots, |\mathrm{Y}|$, perform the following two steps:

(3) Add $\widehat{\mathrm{D}}_t$ to $\widetilde{\mathrm{D}} = \widetilde{\mathrm{D}} \cup \widehat{\mathrm{D}}_t$

(4) Continue using $\widetilde{\mathrm{D}}$ to jointly update $\hat{y}$ and $\mathrm{F}_{\theta}(\cdot)$ over T cycles.

For the joint update in step 4, given a merged dataset $\widetilde{\mathrm{D}}$, PL-ERC applies formula (3) to train y and θ:

$$\mathrm{L}\left(\widetilde{\mathrm{D}}; \hat{y}, \theta\right) = -\frac{1}{|\widetilde{\mathrm{D}}| M} \sum_{i=1}^{|\widetilde{\mathrm{D}}|} \sum_{j=1}^{M} \left(1 - p_{ij}\right)^{\gamma} \cdot \ell_{ce}\left(p_{ij}, \hat{y}_{ij}\right) \tag{3}$$

where ℓ_{ce} is the cross-entropy loss; due to the severe imbalance in sentiment proportions in dialogue sentiment analysis, this work introduces a hyperparameter γ to mitigate data imbalance.

Regarding the parameter θ, PL-ERC can be updated directly using gradient-based methods. For each $\hat{y}_{ij}$, it is updated by improving the current prediction p_{ij} using formula (4):

$$\hat{y}_{ij} = \begin{cases} \dfrac{\hat{p}_{ij} \circ y_i}{\left|\hat{p}_{ij} \circ y_i\right|_1}, & \textit{if } \max\left(\dfrac{\left|\hat{p}_{ij} \circ y_i\right|_1}{\left|\hat{p}_{ij} \circ y_i\right|_1}\right) > \alpha \\ \hat{y}_{ij}, & \textit{otherwise} \end{cases} \tag{4}$$

Here, ∘ denotes element-wise multiplication; $|\cdot|_1$ is the ℓ_1 norm; $\max(\cdot)$ denotes the maximum value of a vector; α is the confidence threshold; $\hat{p}_{ij}$ is the sharpened version of p_{ij} calculated using the temperature parameter τ, as shown in formula (5) below:

$$\hat{p}_{ij} = \frac{p_{ij}^{1/\tau}}{\sum_{k=1}^{|Y|} p_{ijk}^{1/\tau}} \tag{5}$$

The update process for $\hat{y}$ includes: (1) each $\hat{y}_{ij}$ must be covered by y_i, and (2) only prediction results p_{ij} with confidence higher than α will be used to update $\hat{y}_{ij}$. In the PL-ERC framework, PL-ERC can apply any existing dialogue sentiment analysis method as the predictor $\mathrm{F}_\theta(\cdot)$.

2.3. Long Short-Term Memory Network (LSTM)

To effectively implement the dialogue sentiment analysis predictor in the PL-ERC framework and gain a deep understanding of its theoretical foundation for processing sequential data, we need a neural network architecture capable of effectively modeling long-range contextual dependencies. Recurrent neural networks (RNNs) and their improved version, long short-term memory networks (LSTMs), are the key technologies for such sequential modeling tasks.

When exploring the theoretical foundation for evaluating text sentiment analysis based on CNN-LSTM networks, understanding the working principles of recurrent neural networks (RNNs) and their improved version, long short-term memory networks (LSTMs), is an indispensable step. RNNs are a type of neural network architecture specifically designed for processing sequential data. Their unique feature lies in their ability to maintain an internal state in the network's hidden layer, theoretically enabling them to capture information from sequences of arbitrary length. This makes RNNs an ideal choice for natural language processing tasks, including text sentiment analysis.

Compared to traditional neural networks, RNNs have memory capabilities, enabling them to retain prior information when processing sequential data. The basic structure of an RNN is shown in Figure 1, which includes an input layer, hidden layer, output layer, and a recurrent unit, allowing information to be transmitted within the network. This grants RNNs a certain degree of temporal awareness when processing sequential data, making them suitable for various fields such as natural language processing, speech recognition, and time series analysis.

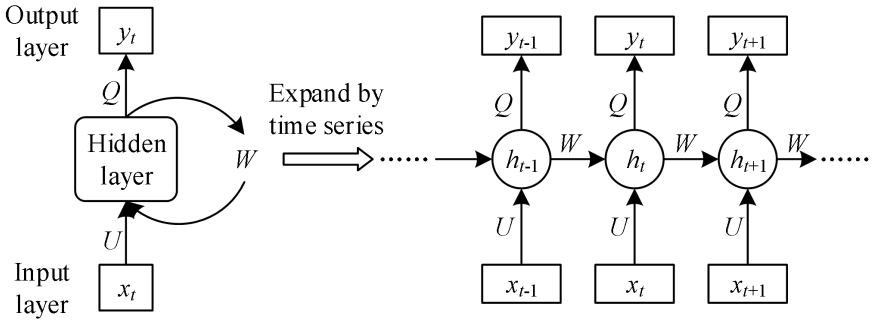

Figure 1. RNN model framework.

The main features of RNNs include:

(1) Recurrent structure: The recurrent structure in RNNs allows information to be passed within the network, enabling the consideration of contextual relationships when processing sequential data.

(2) Weight sharing: In RNNs, the same weights are used for each time step, allowing the network to share parameters across different time steps and reducing the number of training parameters.

(3) Memory capacity: RNNs have a certain memory capacity, enabling them to retain previous information when processing sequential data, which is crucial for understanding context.

Although RNNs perform well in certain tasks, they also have some issues, such as difficulty in capturing long-range dependencies and gradient vanishing or gradient explosion problems during training. To address these issues, several improved RNN structures have been proposed, including Long

Short-Term Memory (LSTM), Bidirectional RNN (Bi-RNN), Bidirectional LSTM (Bi-LSTM), and Gated Recurrent Unit (GRU).

Among these, LSTM is a special type of RNN that can capture long-range dependencies and overcome the limitations of standard RNNs. The core of LSTM is to use memory cells to remember long-term historical information and regulate the flow of information by introducing a set of structures called “gates,” which include: forget gates, input gates, and output gates. This design enables LSTM to effectively store and access information in long sequences, greatly improving the model's ability to handle long-term dependency problems.

The various gates and memory cell units in the “gate” mechanism are shown in Figure 2, with their expressions as follows.

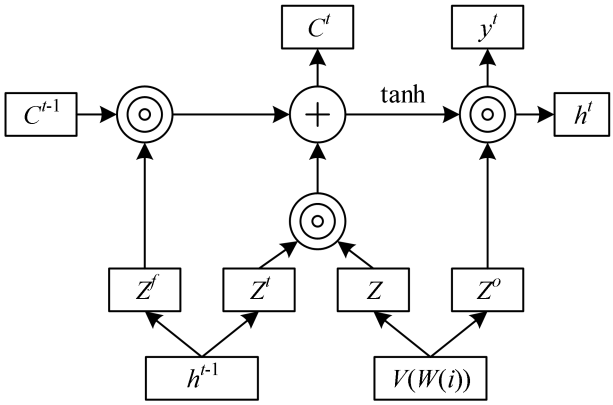

Figure 2. LSTM unit structure.

The LSTM forget gate determines how much of the previous cell state $t-1$ should be retained. The calculation expression is as follows:

$$Z^f = sig\,\mathrm{mod}\left(W_f * [V(W(i), h^{t-1}] + b_f\right) \tag{6}$$

The LSTM input gate determines how much of the current input i should be updated to the cell state, calculated by the expression:

$$Z^i = sig\,\mathrm{mod}\left(W_i * [V(W(i), h^{t-1}] + b_f\right) \tag{7}$$

Calculation of LSTM unit candidate values, which generates a new candidate value to update the unit status. Calculation expression:

$$Z = \tanh\left(W_c * [V(W(i), h^{t-1}] + b_f\right) \tag{8}$$

The update of the LSTM cell unit state combines the results of the forget gate and input gate to update the unit state. Expression:

$$C^t = Z^f * C^{t-1} + Z^i * Z \tag{9}$$

The calculation of the LSTM output gate determines how much of the current unit state C^t should be output to the hidden state. Calculation expression:

$$Z^o = sigmoid\left(W_o * [V(W(i), h^{t-1}] + b_o\right) \tag{10}$$

The final output of LSTM, the update of the hidden state, combines the output gate and the current cell state to produce the output or hidden state of the current time step. Calculation expression:

$$h^t = Z^o * \tanh\left(C^t\right) \tag{11}$$

The symbols in the formula are as follows:

i : Input at the current time step.

h^{t-1} : Hidden state at the previous time step.

c^{t-1} : Cell state at the previous time step (also called cell state).

W : Weight matrix, where the subscripts f,i,c,o correspond to the forget gate, input gate, cell candidate, and output gate, respectively.

b : Bias term, subscripts f,i,c,o are the same as above.

Z^f, Z^i, Z^c, Z^o : Intermediate variables of the gated unit and cell candidate.

σ : Sigmoid activation function, used to output values between 0 and 1.

$\tanh$: Hyperbolic tangent activation function, used to output values between -1 and 1.

In this study, the application of LSTM is mainly focused on utilizing its powerful sequence data processing capabilities to capture time-dependent relationships in text, i.e., how to understand the true meaning of each word or phrase based on contextual information. In sentiment analysis tasks, this means that LSTM can better understand the overall structure and emotional flow of a sentence, even if the emotional expression is extended or indirectly expressed.

2. Cross-Domain Low-Annotation Task-Oriented Dialogue Experiment

Based on the proposed PL-ERC framework and LSTM modeling capabilities, Chapter 3 designs systematic experiments to validate its practical effectiveness in cross-domain, sparsely labeled scenarios. By comparing the performance of mainstream baseline models in multi-domain sentiment analysis tasks for literary works, this method demonstrates its transfer advantages under data-scarce conditions and provides technical support for subsequent fine-grained sentiment analysis.

3.1. Cross-Domain Low-Labeling Scenario Experiment

3.1.1. Dataset

To validate the practical application performance of the proposed progressive learning dialogue-level annotation-based sentiment analysis model in cross-domain few-shot scenarios, experiments were conducted on the SNIPS dataset to evaluate the model's ability to transfer from data-rich domains to unseen few-shot domains. Based on the literary work selection criteria outlined earlier, the paper constructed a 10-shot SNIPS dataset. This dataset includes seven domains with different label sets: Mystery (M), Romance (L), History (H), Biography (B), Science Fiction (S), Children's (C), and Drama (D) literature. Each domain contains 150 small-sample sets, with each small-sample set consisting of a support set and a query set.

3.1.2. Experimental Setup

This experiment was conducted in a cross-domain setting with a 10-shot few-shot configuration to evaluate the model's ability to transfer from data-rich domains to unseen few-shot domains. For the method proposed in this chapter, the same minimal GPT-2 model was used as the base model without introducing new parameters. Pre-training was performed in the source domain, followed by fine-tuning on the target few-shot domain. The learning rate is set to 0.001, with a batch size of 32 during pre-training and a batch size of 2 during 10-shot fine-tuning. During fine-tuning, the same AdamW optimizer and linear decay scheduler are used. Hyperparameters are determined based on performance on the development set.

3.1.3. Comparison with Baseline Model

Here are some competitive benchmark methods, including traditional fine-tuning methods and advanced few-shot learning methods.

Bi-LSTM: Bidirectional Long Short-Term Memory Network, which uses GLoVe word embeddings and bidirectional long short-term memory networks for slot labeling and is trained on the support set.

SimBERT: Similarity BERT, a metric-based method that uses the cosine similarity of BERT embeddings to assign labels to the most similar tokens.

MN: Matching Network is a few-shot sequence labeling model based on matching networks, using BERT as word embeddings.

TransferBERT: Transfer BERT is a conventional NER model based on domain transfer, pre-trained with BERT and then fine-tuned on the target domain support set.

WPZ: A metric-based few-shot slot labeling method similar to MN, but based on prototype networks, which classify by calculating the similarity to the center point of each label sample.

Task-Adaptive Network + Folded Label Transfer (TapNet+CDT), Label-Enhanced Task-Adaptive Network + Folded Label Transfer (L-TapNet+CDT), Label-enhanced Prototype Network + Folded Label Transfer (L-WPZ+CDT) are metric-based few-shot learning methods specifically designed for slot

labeling. They introduce a CRF-based framework to consider the relationships between different slots and a folded label transfer mechanism to learn the transfer probabilities between sequences of different label types.

ConVEx is a fine-tuning-based method that models slot labeling as a fill-in-the-blank task, first pre-training on Reddit data and then fine-tuning on few-shot slot labeling data.

3.1.4. Experimental Results

This experiment uses the F1 evaluation metric to describe the performance of each model in cross-domain few-shot scenarios. Table 1 shows the results of the cross-domain few-shot settings.

Table 1. The F1 value results of each model on cross-domain few-shot applications.

	M	L	H	B	S	C	D	Average
Bi-LSTM	24.56	36.53	34.13	35.51	23.98	44.27	13.66	30.38
SimBERT	33.67	43.09	56.56	57.19	44.48	61.74	42.62	48.48
MN	34.54	49.48	45.12	49.61	37.52	52.65	60.64	47.08
TransferBERT	64.42	62.85	51.69	55.96	62.93	75.17	53.15	60.88
WPZ	59.42	61.47	60.42	74.22	67.64	79.47	69.61	67.46
TapNet+CDT	73.22	74.71	65.46	80.01	77.02	71.87	61.25	71.93
L-TapNet+CDT	83.59	75.11	66.19	74.43	73.61	83.96	71.34	75.46
L-WPZ+CDT	77.78	84.58	74.13	70.82	76.85	87.75	72.17	77.73
ConVEx	80.65	93.99	87.89	81.46	85.89	89.46	82.47	85.97
OURS	92.06	91.08	85.34	82.95	88.57	95.68	86.41	88.87

Table 1 shows the F1 scores of different models in a 10-shot cross-domain sentiment analysis task across seven categories of literary works. Experimental data show that the Bi-LSTM model, as the baseline model, achieves an average F1 score of only 30.38%, with the worst performance in dramatic works (13.66%), indicating its difficulty in handling complex contexts. Measurement learning methods such as SimBERT and MN achieve average F1 scores of 47.08%–48.48%, still significantly lower than fine-tuned models. Models incorporating CRF structures (e.g., TapNet+CDT, L-TapNet+CDT) achieved average F1 scores of 71.93%–75.46%, demonstrating that sequence modeling can effectively enhance performance. L-WPZ+CDT performed exceptionally well in romance-themed works (84.58%), but weaker in the biography category (70.82%), indicating domain sensitivity. ConVEx achieves an average F1 score of 85.97% through a pre-training + fine-tuning strategy, particularly reaching 93.99% in the romance category.

The proposed method leads in almost all aspects, except for romance and history genres, where it achieves 91.08% and 85.34%, respectively, slightly lower than ConVEx. However, it ranks first in other genres such as mystery, children's, and drama, with an average F1 score of 88.87%, surpassing the second-best ConVEx by 2.9%. The model's inter-domain variation is only 12.73% (from 86.41% in the drama category to 95.68% in the children's category), demonstrating stability and lower variability than the comparison models.

3.2. Ablation Experiment

For scenarios with no source domain transfer and few labels, experiments were conducted with $K \in \{10, 50, 100, 200, 500\}$. Ablation experiments were performed from two aspects: adopting a self-training strategy and replacing the improved long short-term memory network (LSTM) with a recurrent neural network (RNN) to comprehensively evaluate the performance of each part of the method proposed in this paper. Detailed analysis was conducted using precision score (P), recall score (R), and F1 score (F).

Table 2 presents the results of the ablation experiments.

Table 2. Analysis of Ablation Experiment Results.

K	Model	P/%	R/%	F1/%
10	PL-ERC-LSTM	88.14	89.61	88.87
	Remove STS	75.49	69.41	72.32
	PL-ERC-RNN	72.55	79.77	75.99
50	PL-ERC-LSTM	90.44	91.19	90.81
	Remove STS	78.84	64.83	71.15
	PL-ERC-RNN	75.75	82.39	78.93
100	PL-ERC-LSTM	91.62	92.34	91.98
	Remove STS	80.44	68.42	73.94
	PL-ERC-RNN	73.49	75.39	74.43
200	PL-ERC-LSTM	92.16	93.25	92.70
	Remove STS	82.86	72.09	77.10
	PL-ERC-RNN	75.71	77.33	76.51
500	PL-ERC-LSTM	95.66	94.43	95.04
	Remove STS	84.18	78.23	81.10
	PL-ERC-RNN	78.53	84.55	81.43

Table 2 shows the performance comparison between the full PL-ERC-LSTM model and two ablation variants under five different annotated data scales. PL-ERC-LSTM maintains the highest F1 score (88.87% → 95.04%) across all data scales and steadily improves with increasing data volume (reaching 95.04% at K=500), with balanced precision and recall rates (e.g., P=92.16%, R=93.25% at K=200), indicating that the model has both accuracy and coverage.

Removing the self-training strategy leads to a significant decline in performance, with the maximum F1 decline reaching 17.72% (at K=50: 90.81% → 71.15%), with recall being particularly affected (R = 64.83% when K = 50 vs. 91.19% for the full model), indicating that the lack of a pseudo-label update mechanism severely weakens context-aware capabilities; the sequence modeling advantages of the LSTM architecture are evident in the PL-ERC-RNN variant, whose F1 scores are consistently lower than those of the LSTM version (maximum difference of 17.99% at K = 100), while RNN exhibits inflated recall rates at low data volumes (R = 79.77% at K = 10) but the lowest precision (72.55%), exposing its limitations in modeling long-range dependencies.

3.3. Source Domain Transfer Target Domain Scenario Experiment

This section evaluates the proposed incremental learning method in cross-domain 1-shot and 5-shot scenarios. In such cross-domain few-shot scenarios, knowledge is transferred from the source domain to an unseen target domain, where the target domain only has a 1-shot or 5-shot support set.

3.3.1. Data Set Settings

Experiments were conducted on two public datasets, Snips and FewJoint. As a popular English dialogue language understanding dataset, the original Snips dataset contains 8 single-intent domains (8 intents) and 57 slots. To form multi-intent domains, single-intent domains were combined, specifically, 4 original domains were combined into training domains, 2 for validation, and 2 for testing.

FewJoint is a Chinese joint dialogue language understanding dataset used in the SMP2020-ECDT task. It contains 62 multi-intent domains, totaling 159 distinct intents and 234 distinct slots. The same FewShot dataset splitting method was directly applied to compare Snips and FewJoint. For Snips, there are 200 few-shot datasets used for training, but these datasets are not used as training data; instead, they are merged for training. The same 50 few-shot datasets are used for development, and 50 few-shot datasets are used for testing. The query set size is 20 (for training and development) and 100 (for testing). Fewjoint is a few-shot task-oriented dialogue understanding benchmark dataset. Following the original data split, there are 50 domains for training, 6 domains for development, and 6 domains for testing.

3.3.2. Evaluation Indicators and Parameter Settings

To maintain consistency with previous work, the slot results from the prompt are converted into sequence-labeled results following the approach outlined in Section 3.1 of this paper. Three metrics were employed in the experiments: Intent Accuracy (IA), Slot F1 Score, and Joint Accuracy (JA). Among these three metrics, Joint Accuracy is considered the most important metric for dialogue language understanding, as it evaluates the accuracy of both intent and slots at the sentence level. For joint accuracy, a sample is considered a positive sample only when all slots and the intent for this sample are correct. The average score across all under-labeled sets is calculated, and the average results of 5 random seeds are reported to control for the uncertainty in neural network training.

The model is similarly fine-tuned on the Snips dataset using the smallest GPT2, fine-tuned on the FewJoint dataset using Chinese GPT2, and uses the AdamW optimizer and linear decay scheduler. Learning rate warm-up and learning rate restart are adopted, with no new parameters introduced. It is important to note that the parameter size of the GPT2 model is nearly identical to that of BERT used in the baseline, ensuring a fair comparison with the baseline. Training was conducted on the training set with a batch size of 32, and fine-tuning was performed on the support set of the validation and test sets with a batch size of 3. During training in the source domain, the intent learning rate and slot learning rate were set to 0.005, and during fine-tuning on the support set of the target domain, both the fine-tuning learning rate for intents and slots was set to 0.007, and the number of iterations during fine-tuning on the support set was set to 50. The above hyperparameters were adjusted based on the joint accuracy performance on the validation set in limited experiments. Other hyperparameters were roughly set based on experience and were not over-tuned.

3.3.3. Experimental Results

Using the baseline model in Section 3.1 for comparison, the performance results of each model on the Snips and FewJoint datasets are shown in Table 3.

Table 3. Performance comparison results on the Snips and FewJoint datasets.

Model	Snips dataset			FewJoint dataset		
	IA/%	F1/%	JA/%	IA/%	F1/%	JA/%
Bi-LSTM	73.71	39.79	6.79	48.35	41.79	14.52
SimBERT	61.44	42.62	9.6	52.29	38.83	11.93
MN	71.22	48.91	18.96	57.37	52.95	26.14
TransferBERT	88.71	51.33	14.65	55.63	50.52	17.26
WPZ	78.67	54.18	12.37	60.08	55.47	16.24
TapNet+CDT	91.47	45.25	15.65	61.61	50.88	21.23
L-TapNet+CDT	82.17	50.83	18.02	64.45	55.18	25.45
L-WPZ+CDT	84.74	54.51	20.83	61.65	52.44	22.63
ConVEx	90.03	57.37	22.93	66.77	61.29	27.29
OURS	94.87	63.93	33.64	70.36	66.25	35.63

The model achieved a joint accuracy (JA) of 33.64% on the Snips dataset, surpassing the second-best model ConVEx's 22.93% by 10.71%; on FewJoint, JA reached 35.63%, outperforming ConVEx by 8.34%. Additionally, the model demonstrates outstanding slot analysis capabilities, achieving an F1 score of 63.93% on Snips (6.56% higher than ConVEx) and 66.25% on FewJoint (4.96% higher than ConVEx). Intent recognition is highly accurate, with IA scores exceeding 94.87% (Snips) and 70.36% (FewJoint) across both datasets. All models achieved lower JA on FewJoint than on Snips, with an average gap of 12.79%. Slot analysis is more challenging in Chinese scenarios, with the best baseline ConVEx achieving F1 = 61.29% vs. 57.37% in English scenarios. The model presented in this paper demonstrates a stronger relative advantage on Chinese datasets, with JA exceeding the second-best model by 8.34% (exceeding the English scenario by 10.71%).

4. Sentiment Analysis Experiment

The cross-domain experiments in Chapter 3 have confirmed the generalization ability of PL-ERC-LSTM, but further exploration of fine-grained sentiment classification is needed for in-depth analysis of literary texts. Chapter 4 focuses on this issue. First, a model is trained, and then a seven-class sentiment analysis comparison experiment is conducted based on a large-scale DMSC dataset to reveal the model's accuracy in capturing complex emotions such as “anger” and “sadness,” laying a technical foundation for educational applications.

4.1. Training Model Evaluation

The model training process for sentiment analysis using a combination of PL-ERC and LSTM models is as follows.

Train the PL-ERC model. Preprocess the data using the jieba library, regular expressions, and a pre-prepared stopword list. Then, use the LineSentence function in the Gensim library to read the processed text file and generate an iterable object, which is used to iteratively train the PL-ERC model.

Train the LSTM model. Randomly divide the entire training data into training, validation, and test sets in an 8:1:1 ratio. Use the training set to enable the LSTM model to learn the relevant patterns in the data and iteratively train the model. During training, it is necessary to provide real-time feedback on the model's performance at each epoch, which is where the validation set comes into play. The test set is used for the final evaluation of the model's performance. The test set consists of data that has never been used for training or validation and can be used to evaluate the model's performance in a real-world environment.

After training, the model's prediction performance across different categories is visualized using metrics such as accuracy, precision, recall, and F1 score, as well as by plotting confusion matrices and accuracy plots. The loss curve and accuracy curve are shown in Figure 3.

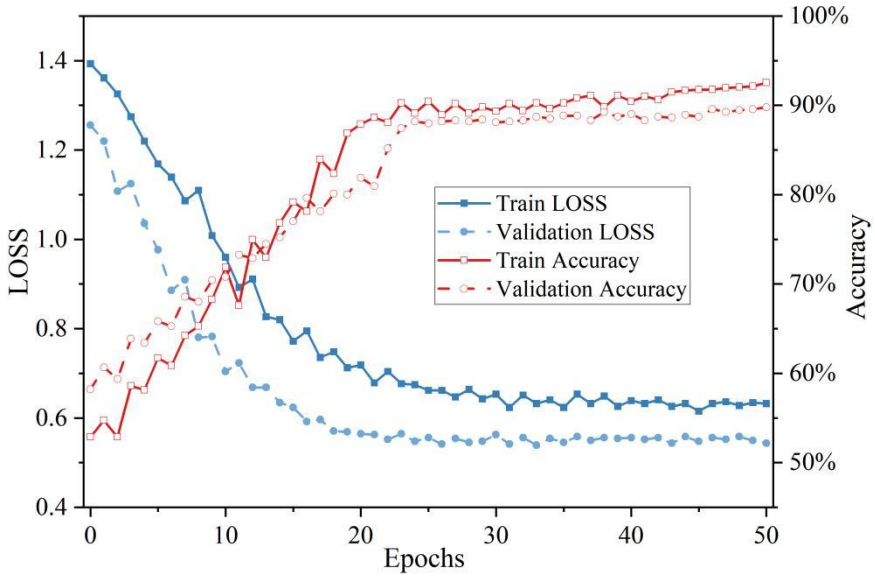

Figure 3. Loss curve and accuracy curve.

As shown in Figure 3, both the training and validation losses decrease during the training process, eventually stabilizing at approximately 25 iterations. The training loss value ultimately remains around 0.63, while the validation loss is slightly lower at 0.54. Similarly, stability is achieved at approximately 25 iterations, with the training set accuracy reaching 92.54% at the 50th iteration, and the validation set accuracy reached 89.78%. This also indicates that the model has been adequately trained.

4.2. Emotion Analysis Comparison Experiment

4.2.1. Experimental Setup and Dataset Description

To validate the performance of the model proposed in this paper for the task of classifying the sentiment of literary works, comparative experiments were conducted on different neural network models using relevant datasets. Each model was tested five times, with a different random seed selected to initialize the model each time. The average of the five experimental results was taken as the final

model performance to reduce the bias in model performance caused by random probability.

To validate the effectiveness of the proposed model, experiments were conducted on the DMSC sentiment classification dataset. The DMSC dataset is a commonly used dataset in the field of text sentiment analysis, containing 381,272 Chinese and foreign literary works, a seven-category fine-grained literary work sentiment analysis dataset, including seven sentiment categories such as happy, angry, and sad.

Since the dataset may have imbalanced category distributions, this paper employs a stratified repeated random subsampling validation method to divide the dataset into training, validation, and test sets in an 8:1:1 ratio, ensuring that the data distribution in each subset remains as consistent as possible with the original dataset's distribution to maintain the accuracy of the experimental results.

4.2.2. Baseline Comparison Model

The selection of comparison models is primarily based on the following perspectives: sequence-based neural network models, attention-based neural network models, graph-based neural network models, and hypergraph-based neural network models.

Sequence-based neural network models:

(1) TextCNN: Text Convolutional Neural Network, which uses convolution and pooling operations to obtain sentiment text representations.

(2) BiLSTM: Bidirectional Long Short-Term Memory Network, which concatenates the last hidden states of both directions to form the global features of the entire text.

Attention-based neural network models:

(3) Transformer: Consisting of an encoder and decoder, the experiment uses only the encoder module to extract text features.

(4) BERT: Obtains sentence vector representations of text based on the BERT pre-training model, then performs classification via a fully connected layer and Softmax.

Graph-based neural network models:

(5) TextGCN: Text Graph Convolutional Neural Network, which converts text classification tasks into node classification tasks by constructing a single heterogeneous graph for text and words, with initial node features using one-hot encoding.

(6) BertGCN: Based on TextGCN, the graph node features are initialized by the pre-trained BERT model, and the two are trained jointly.

(7) TextFCG: A graph neural network that integrates contextual information, constructs a graph for each text, and generates diverse edge representations by combining rich semantic associations to improve the connectivity of the graph and enhance the expressive power of the graph neural network.

Hypergraph-based neural network models:

(8) HyperGAT: A hypergraph attention network that constructs a hypergraph for each text, using an attention mechanism to aggregate information between hyperedges and hypernodes, and converts it into a graph classification task. In the construction of the hypergraph, sequential hyperedges are constructed based on sentences, and LDA topic models are used to construct topic hyperedges. Initial node features are encoded using one-hot encoding.

(9) IBHC: Constructs a sequential hyperedge hypergraph, using spectral hypergraph convolutional networks and BERT to extract text features, which are then combined via an attention mechanism.

4.2.3 Analysis of Experimental Results

Table 4 shows the experimental results of the model in this paper and the above comparison models on the DMSC dataset. This paper selects accuracy, precision, recall, and Micro-F1 value as evaluation indicators.

Table 4. The experimental results of each model on the DMSC dataset.

Model	Accuracy/%	Precision/%	Recall/%	F1/%
TextCNN	93.12	90.85	87.72	89.26
BiLSTM	90.13	94.36	90.51	92.39
Transformer	88.48	87.39	86.45	86.92

BERT	85.33	92.14	85.20	88.53
TextGCN	92.82	93.77	89.84	91.76
BertGCN	94.85	93.54	94.60	94.07
TextFCG	93.73	94.42	92.02	93.20
HyperGAT	94.14	92.82	91.45	92.13
IBHC	95.19	93.38	90.85	92.10
PL-ERC-LSTM	97.17	95.58	96.50	96.04

Table 4 compares the performance of various neural network models in the task of sentiment analysis of literary works. The PL-ERC-LSTM model significantly outperforms all baseline models with an accuracy rate of 97.17%, which is nearly 2 percentage points higher than the second-best model, IBHC (95.19%). Its F1 score (96.04%) is also the highest, improving by 1.97% over the next-best model, BertGCN (94.07%), thereby validating the effectiveness of combining the progressive learning framework with LSTM. The advantage of word sequence models is evident, with sequence models such as BiLSTM (F1 92.39%) and TextCNN (F1 89.26%) outperforming pure attention models (e.g., Transformer's F1 is only 86.92%), indicating that temporal modeling is crucial for sentiment analysis. BERT (F1 88.53%) did not perform as expected, possibly due to the long-context dependencies in literary works exceeding the processing capabilities of standard pre-trained models. Graph neural network-based models (e.g., BertGCN, TextFCG) generally performed well, with BertGCN achieving a recall rate of 94.60%, demonstrating that graph structures can effectively capture textual semantic relationships. However, hypergraph models (HyperGAT, IBHC) had high accuracy (94.14%-95.19%) but lower-than-expected F1 scores (92.10%-92.13%), suggesting that hyperedge construction may introduce noise.

4.3. Classification Confusion Matrix

In order to analyze the performance of the model in this paper across different sentiment categories, a sentiment classification confusion matrix corresponding to the dataset was plotted based on the experimental results, as shown in Figure 4.

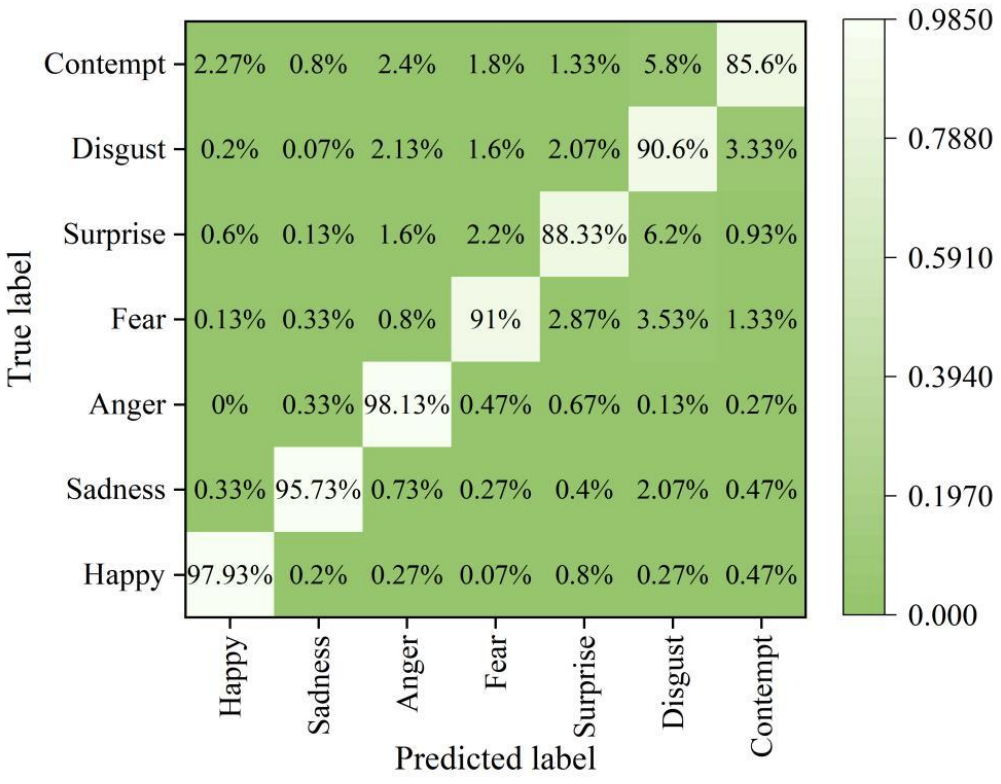

Figure 4. Emotion classification confusion matrix.

Figure 4 shows the fine-grained classification performance of PL-ERC-LSTM for seven categories of emotions. The model is most accurate in identifying anger (98.13%) and happiness (97.93%), while contempt (85.60%) and fear (91.00%) are more difficult to classify. The surprise category has the highest misclassification rate (11.67%), with 6.20% incorrectly classified as disgust, indicating that the model lacks sufficient sensitivity to the contextual differences between "surprise" and "disgust." The disgust category was easily confused with contempt (5.80%), but its recall rate reached 90.60%, indicating that while the model could detect disgust, it struggled to precisely distinguish its subtypes. The anger category had an error rate of only 1.87%, with misclassifications distributed evenly (with the highest being contempt at 2.40%), suggesting that high-intensity emotional features are easier to capture. The recall rate for sadness is 95.73%, with only 0.80% misclassified as contempt, demonstrating the model's robust ability to identify depressive emotions. Overall, the model performs nearly perfectly on basic emotions (happiness/anger/sadness), but the segmentation of complex emotions such as fear and contempt remains a challenge.

5. Teaching Applications Based on Emotional Analysis of Literary Works

To validate the practical application effectiveness of the proposed progressive learning PL-ERC-LSTM dialogue sentiment model in actual literary and cultural education, an actual teaching model was implemented and evaluated at a certain university. The study focused on students majoring in Chinese Language and Literature from the Class of 2024 at a certain university, selecting two classes totaling 86 students. The experimental class (43 students) applied the literary work sentiment analysis method proposed in this paper, while the control class (43 students) used traditional teaching methods, initiating a semester-long comparative teaching experiment.

Prior to the experiment, pre-tests were conducted on students from both classes. The results showed no significant differences between the two classes in terms of self-learning ability, collaborative ability, problem-solving ability, and innovative ability, indicating that the comparative experiment could proceed.

5.1. Analysis of Student Performance in the Teaching Process

After a semester of experimental teaching, teachers reviewed and summarized the knowledge points covered in the learning process, reflected on and summarized the shortcomings that arose during the teaching process, and statistically analyzed the students' overall performance results. Table 5 shows the analysis of student performance for the two classes during the teaching process.

Table 5. Analysis of students' academic performance in two classes.

Evaluation aspect		Experimental Class	Control class
Learning process (20%)	Student self-assessment (5%)	92.94±3.24	90.59±4.29
	Teacher evaluation (15%)	95.92±2.19	84.28±7.28
Report results (20%)	Student self-assessment (5%)	93.71±3.01	92.54±5.18
	Teacher evaluation (15%)	96.71±2.11	86.61±8.45
Post-event summary (20%)	Student self-assessment (5%)	97.88±1.08	94.54±2.98
	Teacher evaluation (15%)	95.39±0.96	78.86±7.29
Final grade (40%)		92.04±5.12	81.63±10.62
Weighted total score		94.25±3.92	84.00±7.89

It can be seen that the experimental class significantly outperformed the control class in all academic evaluation dimensions. The weighted average score of the experimental class reached 94.25 ± 3.92,

significantly higher than that of the control class (84.00 ± 7.89), with a score difference of over 10 points and a lower standard deviation, indicating that the experimental class's performance was more stable. During the learning process, the experimental class's teacher scores were 95.92 ± 2.19, far higher than those of the control class (84.28 ± 7.28), with a score difference of nearly 12 points. In terms of reporting outcomes, the experimental class scored 96.71 ± 2.11, while the control class scored only 86.61 ± 8.45, with a difference of over 10 points. In terms of post-course summaries, the experimental class scored 95.39 ± 0.96, while the control class scored 78.86 ± 7.29, with a difference of 16.53 points. The experimental class's final exam scores were 92.04 ± 5.12, while the control class's scores were 81.63 ± 10.62, with a difference of over 10 points and a lower standard deviation in the experimental class. The emotion-based teaching method significantly improved academic performance, particularly in teacher-led evaluation components (such as presentation of results and final summaries), and notably reduced fluctuations in student performance.

5.2. Analysis of Student Questionnaire Survey on Teaching Effectiveness

Following the conclusion of the teaching experiment, an immediate evaluation of the learning outcomes of students in the experimental and control groups was conducted. Scores from various evaluation scales throughout the learning process were statistically analyzed to serve as indicators for assessing the experimental teaching effectiveness of the literary work emotional analysis method proposed in this paper within the context of literary and cultural education.

Surveys were conducted to assess students' text interpretation abilities, cultural knowledge reserves, critical thinking abilities, writing abilities, and aesthetic experience abilities. Scores from the evaluation scales were statistically analyzed at various stages of the activity (5 points for “very good,” 4 points for “good,” 3 points for “average,” 2 points for “poor,” and 1 point for “very poor”), yielding the results shown in Table 6.

Table 6. Analysis of the questionnaire survey on teaching effect by students.

Dimension	Class	Number of students					Average score
		5	4	3	2	1	
Text interpretation ability	Experimental Class	35	4	3	1	0	4.70
	Control class	14	20	4	4	1	3.98
Reserve of cultural common sense	Experimental Class	32	5	6	0	0	4.60
	Control class	16	15	4	6	2	3.86
Critical thinking ability	Experimental Class	35	4	2	2	0	4.67
	Control class	9	13	14	5	2	3.51
Writing ability	Experimental Class	30	6	4	2	1	4.44
	Control class	6	12	17	6	2	3.33
Aesthetic experience ability	Experimental Class	34	7	1	1	0	4.72
	Control class	14	9	16	3	1	3.74

Students in the experimental class scored significantly higher than those in the control class in all five core competencies, with the largest gap observed in critical thinking ability. The experimental class averaged 4.67 points, while the control class scored only 3.51 points (a difference of 1.16 points). The experimental class also performed best in aesthetic experience ability, averaging 4.72 points (with 34 students scoring full marks), while the control class averaged 3.74 points. The control class performed weakest in writing ability (3.33 points) and critical thinking ability (3.51 points), with over 50% of students scoring ≤3 points. The lowest score in the experimental class was for writing ability (4.44

points), which was still significantly higher than the highest score in the control class (text interpretation at 3.98 points).

To more clearly illustrate the score distributions of the two classes across these five dimensions, scatter plots of the two classes are shown in Figures 5 and 6, respectively. It can be seen that over 80% of students in the experimental class scored 4–5 points (e.g., text interpretation ability: 39/43 students scored ≥4 points), while only 30–60% of students in the control class scored high marks. The experimental class had no 1-point evaluations, while the control class had 1-2 point evaluations in each category.

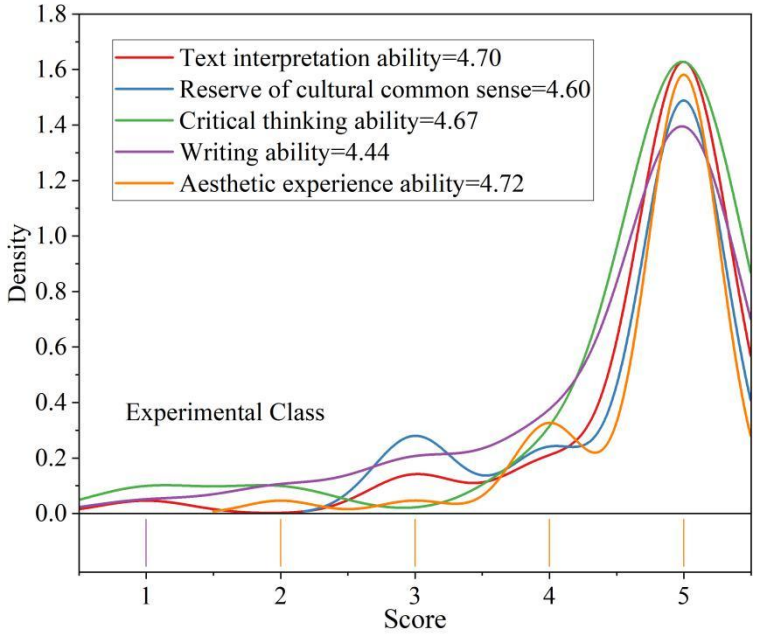

Figure 5. Distribution map of teaching effects of students of the experimental class

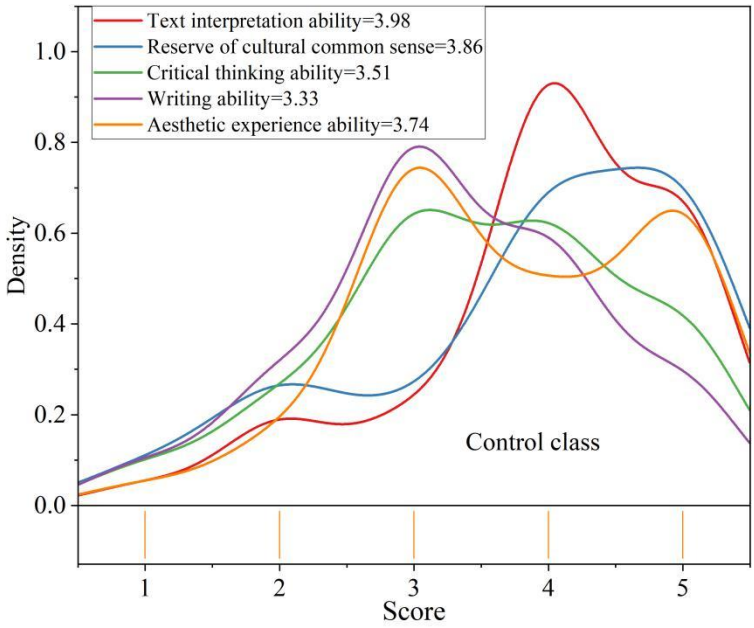

Figure 6. Distribution map of teaching effects of students in the control class

Overall, sentiment analysis methods significantly improved students' higher-order skills (such as critical thinking and aesthetic experience), with the most notable improvements seen in areas that are traditionally weak in traditional teaching (such as writing and critical thinking).

6. Conclusion

This study addresses the challenges of insufficient granularity and scarce annotations in literary work sentiment analysis by proposing an innovative solution that integrates the progressive learning framework PL-ERC with LSTM temporal modeling. Through systematic experiments and educational validation, the following conclusions were drawn:

(1) In cross-domain 10-shot tasks, the average F1 score reached 88.87%, an improvement of 2.9% over the best baseline.

(2) The LSTM gating mechanism significantly enhances long-range dependency modeling, with ablation experiments showing a contribution rate of 12.88%.

(3) In the DMSC seven-classification task with 381K samples, the overall accuracy is 97.17%, and the Micro-F1 is 96.04%, outperforming the second-best model (IBHC 95.19%) by 1.98%.

(4) High-intensity emotion recognition stands out, with anger (98.13%) and happiness (97.93%) recall rates leading the pack. Contempt (85.60%) and fear (91.00%) classification remain challenging and require further optimization of context awareness.

(5) A comparative experiment with a Chinese language and literature major at a university confirmed that the experimental class's weighted total score was 94.25 ± 3.92, significantly higher than the control class (84.00 ± 7.89), with a 50.4% reduction in standard deviation. The experimental class scored higher than the control class in critical thinking (4.67/5) and aesthetic experience (4.72/5), with the largest gap reaching 33.0%.

References

1. Pei, G., Li, H., Lu, Y., Wang, Y., Hua, S., & Li, T. (2024). Affective computing: Recent advances, challenges, and future trends. Intelligent Computing, 3, 0076.
2. Politou, E., Alepis, E., & Patsakis, C. (2017). A survey on mobile affective computing. Computer Science Review, 25, 79-100.
3. Chan, J. Y. L., Bea, K. T., Leow, S. M. H., Phoong, S. W., & Cheng, W. K. (2023). State of the art: a review of sentiment analysis based on sequential transfer learning. Artificial Intelligence Review, 56(1), 749-780.
4. Ligthart, A., Catal, C., & Tekinerdogan, B. (2021). Systematic reviews in sentiment analysis: a tertiary study. Artificial intelligence review, 1-57.
5. Yadollahi, A., Shahraki, A. G., & Zaiane, O. R. (2017). Current state of text sentiment analysis from opinion to emotion mining. ACM Computing Surveys (CSUR), 50(2), 1-33.
6. Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., & Zhao, L. (2019). Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. Multimedia tools and applications, 78, 15169-15211.
7. Zimbra, D., Abbasi, A., Zeng, D., & Chen, H. (2018). The state-of-the-art in Twitter sentiment analysis: A review and benchmark evaluation. ACM Transactions on Management Information Systems (TMIS), 9(2), 1-29.
8. Alamoodi, A. H., Zaidan, B. B., Zaidan, A. A., Albahri, O. S., Mohammed, K. I., Malik, R. Q., ... & Alaa, M. (2021). Sentiment analysis and its applications in fighting COVID-19 and infectious diseases: A systematic review. Expert systems with applications, 167, 114155.
9. Araque, O., Corcuera-Platas, I., Sánchez-Rada, J. F., & Iglesias, C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. Expert Systems with Applications, 77, 236-246.
10. Peng, H., Ma, Y., Li, Y., & Cambria, E. (2018). Learning multi-grained aspect target sequence for Chinese sentiment analysis. Knowledge-Based Systems, 148, 167-176.
11. Yun Ying, S., Keikhosrokiani, P., & Pourya Asl, M. (2022). Opinion mining on Viet Thanh Nguyen's the sympathizer using topic modelling and sentiment analysis. Journal of Information Technology Management, 14(Special Issue: 5th International Conference of Reliable Information and Communication Technology (IRICT 2020)), 163-183.
12. Al-Moslmi, T., Omar, N., Abdullah, S., & Albared, M. (2017). Approaches to cross-domain sentiment analysis: A systematic literature review. Ieee access, 5, 16173-16192.
13. Klinger, R., Suliya, S. S., & Reiter, N. (2011). Automatic Emotion Detection for Quantitative Literary Studies: A case study based on Franz Kafka's "Das Schloss" und "Amerika". development, 165.
14. Park, M., Park, S., & Shin, H. (2022, January). Literature Representation using Character Networks based on Sentiment Analysis. In 2022 IEEE International Conference on Big Data and Smart Computing (BigComp) (pp. 190-193). IEEE.

15. Labatut, V., & Bost, X. (2019). Extraction and analysis of fictional character networks: A survey. ACM Computing Surveys (CSUR), 52(5), 1-40.
16. Chu, K. E., Keikhosrokiani, P., & Asl, M. P. (2022). A topic modeling and sentiment analysis model for detection and visualization of themes in literary texts. Pertanika Journal of Science & Technology, 30(4), 2535-2561.
17. PM, K. R. (2022). Sentiment analysis, opinion mining and topic modelling of epics and novels using machine learning techniques. Materials Today: Proceedings, 51, 576-584.
18. Klinger, R., Kim, E., & Padó, S. (2020). Emotion Analysis for Literary Studies. Reflektierte Algorithmische Textanalyse: Interdisziplinäre (s) Arbeiten in der Creta-Werkstatt. De Gruyter, 237-268.
19. Yuri, B., & Pascale, F. (2024). Sentiment Analysis for Literary Texts: Hemingway as a Case-study. Journal of Data Mining & Digital Humanities.