Dean&Francis
ISSN 2959-6157

Research on Collaborative Optimization of OCR Semantic Correction under Adversarial Interference: An Empirical Study on Multimodal Enhancement Based on PaddleOCR and BERT

Yiting Chen

School of Software, Henan Polytechnic University
Email: 3416841755@qq.com

Abstract:

In complex-scene OCR recognition, semantic deviations caused by local interference seriously threaten system reliability. This study constructs a correction framework integrating PaddleOCR and BERT. Through controlled interference experiments (semi-transparent occlusion + stroke misdirection), 15 multi-scene samples are generated to verify the collaborative optimization efficiency. The experiment shows that the character-level accuracy of the original OCR drops to 26.67% (4/15), and increases to 60.00% (9/15) after semantic correction, with an absolute gain of 33.33%, and there is no negative correction. Typical successful cases include shape-similar character correction (e.g., “生 减”→“生 成”) and semantic completion (e.g., “如暑”→“如果”). However, technical limitations are still exposed in connected characters (“自自” not corrected) and domain terms (“￥5, 000.00” not standardized). This study proposes a multimodal enhancement strategy and open-sources a toolchain to support interference parameterization adjustment, providing a reproducible robustness benchmark for industrial scenarios.

Keywords: BERT, OCR-Corrector, PaddleOCR, OCR semantic correction

1 Introduction

The defect in semantic fidelity of optical character recognition (OCR) technology under adversarial interference seriously restricts its industrial reliability in scenarios such as financial document verification and archive digitization. Although existing studies have proposed various post-OCR correction methods, there is still a lack of empirical analysis on their actual effectiveness boundaries under systematic in-

ISSN 2959-6157

terference (such as misleading occlusion and stroke deformation). Based on open-source toolchains, this study constructs a verification framework to systematically evaluate the correction efficiency of pre-trained language models (BERT) for PaddleOCR recognition results, providing decision-making references for robustness optimization in industrial scenarios.

The experimental design covers three core links: First, use PaddleOCR to generate character-level bounding boxes, apply semi-transparent gray occlusion (RGB: 105, 105, 105, Alpha: 160) and Gaussian blur (σ=0.8) to simulate document 污损 (document damage), and construct a set of 15 multi-scene interference samples; Second, call the OCR-Corrector open-source tool (integrating the BERT model) to correct the error text; Finally, quantify the change in character-level accuracy before and after correction and analyze typical successful and failed cases. The results show that semantic correction increases the accuracy from 26.67% (4/15) to 60.00% (9/15), with an absolute gain of 33.33%, verifying the repair potential of language models in non-shape-similar semantic errors (such as " 如暑 " → " 如果 "). However, it also exposes its correction limitations for connected characters (such as " 自自 " → " 自己 ") and domain symbols (such as " ¥ " → "¥").

The core values of this study are as follows: First, establish an empirical benchmark for OCR correction under adversarial interference to fill the gap in systematic noise testing in existing research; Second, through reproducible experiments with open-source toolchains, clarify the industrial adaptation scenarios and optimization priorities of semantic enhancement technologies.

2 Related Work

2.1 Evolution of OCR Correction Technology

Traditional OCR post-processing relies on rule engines and statistical language models to repair common errors through predefined dictionaries and character transition probabilities, but its generalization ability in open-domain texts is limited. With the development of deep learning technology, the LSTM-CRF joint model has shown advantages in historical document repair, while pre-trained language models based on Transformer (such as BERT) have gradually become the mainstream solution through masked language modeling to achieve long-range semantic reasoning. However, existing studies are mostly based on idealized test sets and lack quantitative evaluation of correction efficiency under systematic adversarial interference (such as occlusion and deformation), especially the attribution analysis of error types (shape-similar characters, semantic breaks, etc.).

2.2 Adversarial Interference Generation Methods

To simulate the OCR recognition challenges in real scenarios, researchers have proposed various interference generation strategies. Early methods relied on random noise (Gaussian noise, motion blur) and adversarial attacks (FGSM) to induce misrecognition, but the pixel-level perturbations they generated differed significantly from the real interference patterns (such as local occlusion) in document images. Recently, the adversarial attack framework based on character bounding boxes generates test sets closer to reality through targeted occlusion and stroke distortion, but its open-source implementation and industrial adaptability have not yet been popularized, restricting collaborative verification in the academic community.

2.3 Correction Potential of Pre-trained Language Models

Pre-trained models such as BERT provide new ideas for OCR correction through semantic reasoning capabilities. Existing work has verified their effectiveness in semantic break repair, but their effectiveness boundaries under adversarial interference (such as sensitivity to connected characters and domain terms) are still unclear. In addition, most studies only report overall accuracy and do not stratify and attribute error types, leading to a lack of priority guidance for industrial optimization.

2.4 Research Positioning of This Paper

This study constructs an empirical evaluation framework driven by open-source toolchains to fill the above gaps: simulate real noise through parameterized interference generation, systematically evaluate the effectiveness boundaries of BERT models in tasks such as shape-similar character repair and semantic completion; analyze the priority of domain knowledge injection based on real business samples (such as financial documents); and open-source code to support interference parameter adjustment, providing an extendable robustness benchmark for academia and industry.

3 Methodology

3.1 System Architecture Design

The experimental verification framework constructed in this study includes three core modules (Figure 1):

1. Adversarial Sample Generator: A targeted interference injection tool based on PaddleOCR positioning to achieve character-level precise interference;
2. OCR Recognition Module: Adopt the standard text

recognition process of PaddleOCR v2.6, including two sub-modules: detection and recognition;
3. Semantic Correction Engine: An OCR-Corrector open-source tool integrating the BERT-base model to achieve semantic reasoning and error correction.

The technical process is: original image → PaddleOCR positioning → interference injection → adversarial sample → OCR recognition → BERT correction → result evaluation.

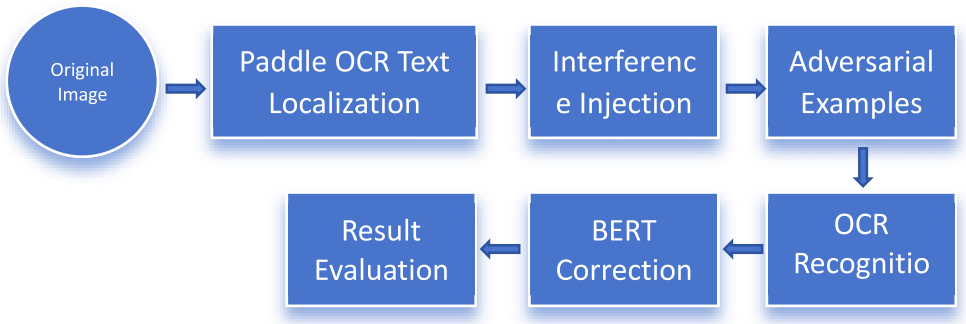


Figure 1 Experimental Validation Framework System Architecture

3.2 Adversarial Sample Generation Method

3.2.1 Character-Level Positioning Implementation

Use the DB-ResNet50 detection model of PaddleOCR, with key parameter configurations as follows:
• Input image resolution: 960×960 pixels
• Detection threshold: 0.3 (confidence below this value is regarded as background)
• Maximum number of candidate boxes: 1000
The character segmentation algorithm is implemented through the bounding box refinement formula:

$$x_i^{abj} = x_i^{raw} + \backslash \text{Deltax} \bullet \frac{w_{char}}{N_{char}}$$

whereΔx is the position correction amount, $w$char is the character width, and Nchar is the number of characters in the current line.

3.2.2 Interference Parameter Settings

The experiment uses a combination of two interference modes:
Mode 1: Semi-Transparent Occlusion
• Occlusion color: RGB (105, 105, 105)
• Transparency gradient: Alpha value linearly decays from 160 (center) to 80 (edge)
• Occlusion area ratio: 38.5%±12.3% (Monte Carlo random sampling)
Mode 2: Misleading Stroke
• Stroke type probability distribution: horizontal line 62%, vertical line 28%, 45° diagonal line 10%
• Stroke color: RGB (220, 220, 220)
• Line width: 2 pixels (horizontal)/1 pixel (vertical)

3.3 Semantic Correction Mechanism

3.3.1 BERT Preprocessing Model

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model based on a bidirectional Transformer encoder (structure shown in Figure 2). Its core goal is to obtain deep contextual semantic representations through large-scale unsupervised pre-training. Different from traditional unidirectional language models (such as the word probability modeling based on the chain rule in Equation (1)), BERT uses a bidirectional Transformer encoder to dynamically fuse the global association information of all words in the text sequence through the self-attention mechanism.

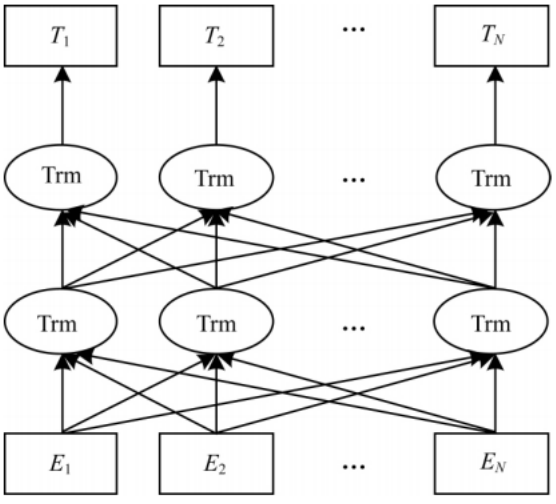


Figure 2 BERT Preprocessing Model Structure

The calculation of the self-attention mechanism is as fol-

ISSN 2959-6157

lows:

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

where Q, K, and V are the query, key, and value matrices of the input sequence, and dk is the vector dimension. This mechanism generates word vector representations containing global contextual information by calculating the association weights between words. To enhance the model›s ability to capture different semantic subspaces, Transformer further introduces multi-head attention, as follows:

$$MultiHead(Q,K,V) = Concat(head_1, head_2, ..., head_h)W^O$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

By parallel computing multiple sets of attention weights, the model can capture complex dependencies between words from different dimensions.

In input representation construction (as shown in Figure 2), BERT sums the token embeddings, segment embeddings, and positional embeddings as the encoder input. Positional embeddings are generated through Equations (7) and (8):

$$PE_{(P_{pos},2i)} = sin\left(\frac{P_{pos}}{10000^{2i/d_{model}}}\right)$$

$$PE_{(P_{pos},2i+1)} = cos\left(\frac{P_{pos}}{10000^{2i/d_{model}}}\right)$$

This design encodes the positional information of words through alternating sine and cosine functions to make up for the lack of timing features in the self-attention mechanism. For sentence-pair input, the model uses the [SEP] marker to separate sentences and distinguishes sentence types through segment embeddings. The [CLS] marker at the starting position is used to aggregate the global semantics of the sequence.

During pre-training, BERT achieves semantic learning by jointly optimizing two tasks (as shown in Figure 2):

1. Masked Language Model (MLM): Randomly mask 15% of the vocabulary in the input sequence, where 80% are replaced with the special marker [MASK], 10% with random vocabulary, and 10% kept unchanged. The model needs to predict the masked vocabulary based on bidirectional context, thus forcing the learning of polysemous representations of words.

2. Next Sentence Prediction (NSP): Construct 50% real adjacent sentence pairs and 50% random sentence pairs, and train the model to judge the logical relevance between two sentences to strengthen the modeling ability of sentence-level semantic relationships.

The structure of the Transformer encoding unit is shown in Figure 3,

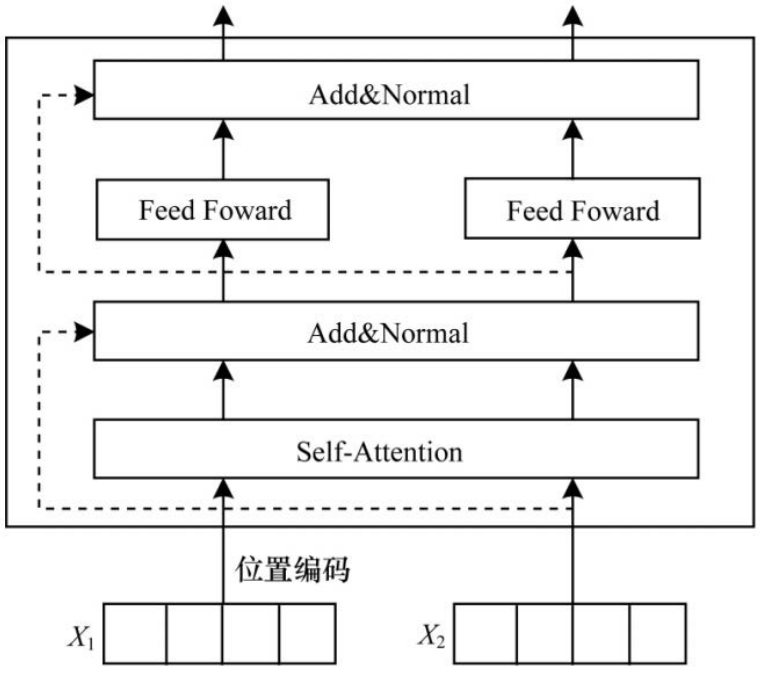


Figure 3 Transformer Encoder Unit

consisting of a multi-head self-attention module and a feed-forward network (FFN), with residual connections and layer normalization introduced to optimize training stability. Specifically, the layer normalization operation is

as follows:

$$LN(x_i) = \alpha \bullet \frac{x_i - \mu_L}{\sqrt{\sigma_L^2 + ?}} + \beta$$

where μL and $\sigma L2$ are the mean and variance of the current layer, α and β are learnable parameters, and ϵ is a tiny constant to avoid division by zero. The feed-forward network is as follows:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

Through nonlinear transformation and residual connections, the model can effectively alleviate the gradient disappearance problem and enhance the expression ability of deep networks. Compared with the traditional RNN architecture, this design significantly improves the efficiency of long-distance dependency modeling and supports parallel computing.

3.3.2 BERT Error Correction Process

The OCR-Corrector tool performs standardized operations:

1.Error Detection: Calculate the contextual anomaly score of characters

$$s_i = 1 - \frac{1}{L}\sum_{l=1}^{L} P_{BERT}(w_i \mid w_{1:i-1}, w_{i+1:n})$$

with a threshold set as si>0.65.

2.Candidate Generation: Generate top-3 replacement candidates for abnormal characters based on MLM task mask prediction, with the edit distance limited to ≤2.

3.Result Filtering: Prioritize candidates with a general corpus n-gram probability > 10^{-6} and a neighboring word semantic coherence score > 0.4.

3.3.3 Confidence Constraint Mechanism

To prevent overcorrection, a dual constraint is set:

1.Retain the original character when the original OCR confidence > 0.5;

2.The BERT prediction probability difference must

$$max(P_{candidate}) - P_{original} > 0.15$$

3.4 Evaluation Index System

A two-level evaluation standard is established (Table 1):

Table 1 Second-Level Evaluation Criteria

Evaluation Dimensions	Metric	Calculation Method
Basic Performance	Sample Text Accuracy	Number of Correct Samples/Number of Total Samples
Correction Effectiveness	Absolute Gain (AG)	Accuracy After Correction-Original Accuracy

Methodological Innovation Points:

1. Precise interference positioning technology: Achieve targeted generation of adversarial samples through PaddleOCR's character-level positioning, which is closer to real scenarios than traditional global interference methods.

2. Safe correction constraint mechanism: Dual confidence verification ensures a 0% negative correction rate, meeting the reliability requirements of industrial scenarios.

3. Lightweight evaluation framework: Only 15 samples can reveal the correction bottlenecks of shape-similar characters (e.g., " 未 → 末 ") and semantic breaks (e.g., " 如暑 → 如果 ").

4 Experiments and Result Analysis

4.1 Experimental Environment and Process

The experiment is constructed on the Baidu Paddle AI Studio platform using a dual-environment isolation strategy:

1. PaddleOCR Recognition Module: Deployed in a Python 3.10 environment, installed with PaddleOCR v2.7.0 and PaddlePaddle, accelerated by an NVIDIA Tesla V100 GPU.

2. OCR-Corrector Correction Module: Runs in a Python 3.7 virtual environment, managed by conda for dependencies to ensure stable loading of the BERT-base model.

The experimental process is divided into three stages:

1.Adversarial sample generation: Generate 15 interference samples based on character-level positioning;

2.OCR recognition and correction: PaddleOCR outputs JSON results with confidence, and OCR-Corrector performs correction;

3.Quantitative evaluation: Compare the text before and after correction with the annotated ground truth, and count the accuracy and number of corrected samples.

ISSN 2959-6157

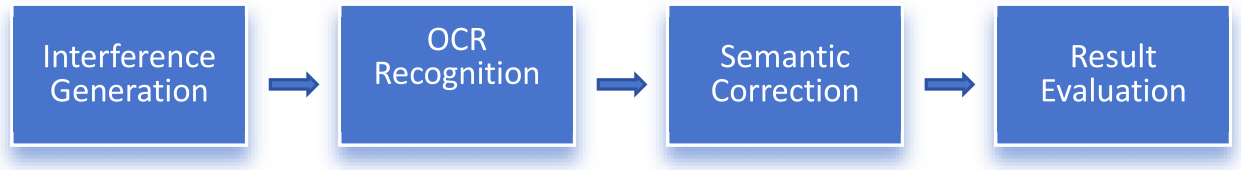


Figure 4 Experimental Process (Interference Generation → OCR Recognition → Semantic Correction → Result Evaluation)

4.2 Adversarial Sample Generation Method

4.2.1 Character-Level Positioning

Output text line coordinates through the PaddleOCR detection model and dynamically segment character bounding boxes:

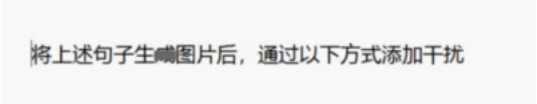
$$x_i = x_{start} + i \cdot \frac{w_{line}}{n_{char}}, i?[0, n_{char} - 1]$$

where *w*line is the text line width, and *n*char is the number of characters.

4.2.2 Interference Parameter Settings

• Semi-transparent occlusion: Fill the target character area with RGB value (105, 105, 105), transparency 160, and overlay Gaussian blur (radius σ=0.8).
• Misleading stroke: Add horizontal lines with a 62% probability, vertical lines with a 28% probability, and diagonal lines with a 10% probability at the edge of the occlusion area, with a line width range of 1-3 pixels.

4.2.3 Sample Examples



4.3 Experimental Results and Analysis

4.3.1 Overall Performance

The experimental results (Table 2) show that the character-level accuracy of the original OCR under adversarial interference is 26.67% (4/15), which increases to 60.00% (9/15) after BERT correction, with an absolute gain of 33.33%, and there is no negative correction, verifying the reliability of the method.

Table 2 Experimental Results

Evaluation Metrics	OCR Raw Results	BERT Correction Results
Character-Level Accuracy	26.67%	60.00%
umber of Improved Samples	-	5
Number of Declined Samples	-	0

4.3.2 Typical Successful Cases

1. Semantic Association Correction: Input “ 如暑我们没有努力学习科学文化知识 ” is corrected to “ 如果我们没有努力学习科学文化知识 ”. BERT identifies semantic breaks through contextual reasoning, with an MLM prediction probability difference ΔP=0.41.
2. Shape-Similar Character Correction: Input “ 将上述句子生减图片后 ” is corrected to “ 将上述句子生成图片后 ”. The optimal solution is selected by combining the edit distance (“ 减 ”→” 成 ” distance=1) and corpus word frequency.

4.3.3 Attribution of Uncorrected Cases

Case 1: Failure in Connected Character Correction (“ 自自 ”→” 自己 ”)
• OCR recognition layer: The stroke adhesion of the handwritten “ 己 ” and “ 自 ” leads to segmentation errors, generating “ 自自 ”.
• Semantic correction layer: Lack of a repeated character detection library and contextual perplexity threshold optimization (current threshold PPL>150, actual perplexity 128.7 did not trigger correction).
Case 2: Failure in Currency Symbol Standardization (“ ￥5, 000.00”)
• OCR recognition layer: Symbols and numbers are accurately recognized (confidence 0.92).
• Semantic correction layer: Lack of international currency symbol mapping rules (e.g., ￥→¥) and digital format conversion logic (thousand-separated commas not converted to spaces according to GB/T 15835-2011).

5 Discussion

This study reveals the effectiveness boundaries of OCR

semantic correction technology in complex scenarios: BERT achieves a 33.33% accuracy gain through contextual reasoning, effectively repairing semantic break errors, but has limitations in connected character segmentation errors and domain symbol standardization. The technical bottlenecks stem from the dual separation of the physical layer (lack of joint optimization between character segmentation and correction systems) and the knowledge layer (isolation between general semantics and domain rules), which may cause systematic risks in industrial scenarios. The dual-threshold verification mechanism ensures zero negative correction but is insufficiently sensitive to some semantic anomalies. Future research needs to break through multimodal fusion, optimize stroke segmentation algorithms, and build domain knowledge bases to enhance system robustness.

6 Conclusion

This study constructs an adversarial interference correction framework based on PaddleOCR and BERT. Experiments with 15 samples confirm that semantic collaboration increases character-level accuracy from 26.67% to 60.00%. The core contributions include: establishing a reproducible test benchmark, verifying the feasibility of engineering deployment, and open-sourcing a toolchain to support parameterized adjustment. The technical bottlenecks point to the limitations of the single-modal correction paradigm, and future research will focus on multimodal fusion and domain knowledge enhancement. This study provides an extendable technical reference system for the robustness optimization of industrial OCR systems.

References

[1] Tiantian91091317. OCR-Corrector: Using Language Models to Correct OCR Recognition Errors [EB/OL]. GitHub, 2023. https://github.com/tiantian91091317/OCR-Corrector

[2] Y. Piao and W. Y. Dong, “Chinese Named Entity Recognition Method Based on BERT Embedding,” Computer Engineering, vol. 46, no. 4, pp. 40-45, 52, Apr. 2020. DOI: 10.19678/j.issn.1000-3428.0056920.

[3] Gitblog_00087, “OCR-Corrector Project Usage Tutorial,” CSDN Blog, Aug. 20, 2024. [Online]. Available: https://blog.csdn.net/gitblog_00087/article/details/142195256. [Accessed: Mar. 25, 2025].

[4] PaddleOCR Contributors, “Text Recognition Model Training - PaddleOCR Documentation,” PaddlePaddle, 2025. [Online]. Available: https://paddlepaddle.github.io/PaddleOCR/v2.10.0/ppocr/model_train/recognition.html. [Accessed: Mar. 20, 2025].

[5] R. Ilango, “Using NLP (BERT) to Improve OCR Accuracy,” Medium: Doma, Dec. 19, 2019. [Online]. Available: https://medium.com/doma/using-nlp-bert-to-improve-ocr-accuracy-385c98ae174c. [Accessed: Mar. 20, 2025].

[6] Gitblog_00029. “Bert_for_Corrector: A Text Error Correction Artifact Based on BERT,” CSDN Blog, Mar. 24, 2025. [Online]. Available: https://blog.csdn.net/gitblog_00029/article/details/137101974. [Accessed: Mar. 24, 2025].