Block Noise Identification Method for Web-Crawled Policy Texts in Enterprise RAG Systems

Huaichuan Yi*

School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, Shaanxi, China
Email: yhc271828@stu.xjtu.edu.cn
*Corresponding author
Manuscript received April 15, 2026; accepted May 17, 2026; published June 15, 2026

Abstract—With the widespread application of Large Language Models (LLMs) in enterprise knowledge base question-answering systems, Retrieval-Augmented Generation (RAG) technology has become a key solution to the hallucination problem of LLMs. However, in web-crawled policy texts, a large amount of block noise—such as navigation blocks, header/footer blocks, advertisement blocks, and semantically incomplete policy fragments—appears superficially identical to normal text, with complete grammatical structures and policy keywords, making it difficult for traditional rule-based methods to identify them. This noise severely degrades the retrieval and generation quality of downstream RAG systems. To address this core challenge, this paper proposes a block noise identification method for web-crawled policy texts. First, a rule-driven pre-admission and cleaning strategy quickly filters out obviously invalid samples; then, a lightweight gating classifier that fuses 32-dimensional handcrafted statistical features with dual-centroid semantic similarity is designed to specifically identify block noise that rules cannot cover. Experimental results show that on a labeled dataset of 4448 text blocks (clean=2315, noisy=2133), the proposed 32-dimensional handcrafted feature baseline achieves an F1 score of 74.01%; the MLP classifier fused with dual-centroid semantic similarity achieves an F1 score of 78.05% (precision 74.11%, recall 82.44%), which is the current best result. This method provides a lightweight, locally deployable engineering solution for the precise identification of block noise in web-crawled texts.

Keywords—RAG; Block Noise Identification; Noise Gating; Semantic Centroid; Text Denoising; Large Language Models

I. Introduction

With the widespread application of Large Language Models (LLMs) in enterprise knowledge base question-answering systems, Retrieval-Augmented Generation (RAG) technology has become a key solution to the hallucination problem of LLMs [1]. In the construction of RAG systems, text preprocessing is the foundational link connecting raw documents and vector retrieval—high-quality preprocessing can remove various types of noise from web-crawled data, preserve valid content, and thereby improve the accuracy of subsequent retrieval and the reliability of generation results [2]. However, when enterprise-level RAG systems process web-crawled policy texts, they face a severely underestimated core challenge: a large amount of block noise in web pages—such as navigation blocks, header/footer blocks, advertisement blocks, and semantically incomplete policy fragments—appears indistinguishable from normal text at first glance. These blocks have complete grammatical structures and policy keywords, making it difficult for traditional methods to distinguish them from genuine policy content. This noise directly contaminates the semantic quality of the knowledge base [3].

This dilemma has clear practical significance in real enterprise applications. Taking the scenario of our collaborating enterprise as an example: it is necessary to batch crawl new energy vehicle policy texts (approximately 400,000 documents) from government public websites, perform OCR recognition followed by preprocessing and denoising, and then build a RAG knowledge base. Existing denoising methods are already quite mature in handling obvious noise (such as HTML tag residues, format corruption, and ultra-short texts), but they expose severe blind spots when facing block noise: navigation bar text such as "New Energy Vehicle Subsidy Policy Overview," footer copyright information such as "This document is for reference only; please refer to the official publication for the authoritative version," and semantically incomplete policy clause fragments—all of these appear to be structurally complete text blocks. As the empirical study by Bevendorff *et al.* [4] demonstrates, existing content extraction algorithms still have limitations when processing complex noise, and rule-based methods struggle to identify them as noise through simple keywords or format features. While directly using cloud-based large models to judge each text block individually could identify such noise, the API call costs are prohibitively high and inference latency is significant, making it difficult to support large-scale processing of 400,000-level documents [5].

The identification of block noise is particularly challenging due to its high similarity to clean text in surface features: both contain policy keywords, both conform to Chinese grammatical norms, and both present as structured text paragraphs. The features relied upon by rule-based methods—HTML tags, fixed keywords—often do not exist in block noise; while pure semantic methods (such as the LSA topic model) can capture semantic deviation, they have limited ability to distinguish boundary samples with fuzzy semantic boundaries (such as incomplete excerpts of policy clauses). Therefore, how to efficiently filter obvious noise while precisely identifying block noise that rules cannot cover, within a single framework, is the core problem this paper aims to solve.

To address the above challenges, this paper proposes two core contributions:

(1) A Two-Stage Denoising Framework for Block Noise Identification. This paper designs a "rule-first, intelligent fallback" two-stage denoising architecture. The first stage quickly filters out obviously invalid samples through rule-based methods (encoding errors, ultra-short

texts, 404 pages, etc.) and pre-filters semantically outlier samples through the LSA topic model, reducing the burden on the second stage; the second stage introduces a lightweight gating classifier specifically designed to identify block noise that rules cannot cover. The core philosophy of this framework is "low-cost first, high-precision fallback": the vast majority of obvious noise is filtered by rule-based methods within millisecond-level latency, and only a small number of suspected block noise samples are escalated to the semantic model for processing.

(2) A Block Noise Gating Classifier Fusing Handcrafted Features and Semantic Centroids. To address the core challenge of high surface feature similarity between block noise and clean text, this paper designs a hybrid classification strategy that fuses 32-dimensional handcrafted statistical features with dual-centroid semantic similarity. First, based on the format features of text blocks (line length, indentation, punctuation density, noise keyword hits, etc.), 32-dimensional handcrafted features are constructed to capture subtle differences in format between block noise; second, the paraphrase-multilingual-MiniLM-L12-v2 model is used to extract semantic vectors, respectively constructing noise semantic centroids and clean semantic centroids to form reverse signals; finally, the dual-centroid similarity is fused with handcrafted features to train a lightweight MLP classifier with only approximately 4480 parameters. Experimental results show that the dual-centroid MLP fused classifier achieves an F1 score of 78.05% (precision 74.11%, recall 82.44%) on a test set of 4448 text blocks, significantly outperforming the 32-dimensional handcrafted feature baseline (+4.04pp), validating the effective identification capability of this classification strategy for block noise.

II. Related Work

A. Web Text Denoising Methods

Existing web text denoising methods can be divided into three categories:

Rule-based methods utilize predefined regular expressions or keyword matching to remove known pattern noise such as HTML tags, advertisement text, and navigation bars [6, 7]. These methods are simple to implement and computationally efficient but cannot handle noise types not covered by rules, and the cost of rule maintenance is high.

Statistical learning-based methods utilize topic models (such as LSA, LDA) or classifiers to identify low-quality samples. Researchers have proposed using TF-IDF combined with LSA topic models for document quality scoring, filtering semantically irrelevant samples through outlier detection [8]. These methods can identify noise at the semantic level but have limited ability to distinguish short texts and boundary samples.

Deep learning-based methods utilize neural networks to automatically learn noise features. Text classification methods based on pre-trained models such as BERT [9] have been widely applied to various text quality assessment tasks, achieving good results. However, these methods have large parameter counts and high inference costs, making them difficult to deploy in resource-constrained enterprise local environments.

B. Noise Classification and Gating Mechanisms

Noise classification is a critical component of text denoising. Existing research primarily divides noise into structured noise (HTML tags, advertisements, navigation) and unstructured noise (OCR errors, semantic incompleteness, garbled characters) [10]. Structured noise is suitable for rule-based processing, while unstructured noise requires semantic understanding capabilities.

The field of text classification has developed a rich system of deep learning models [11], and dynamic neural networks have been widely applied in NLP tasks, with the core idea being to use lightweight models to quickly filter simple samples, only delegating difficult samples to complex models [12]. This paper applies this idea to the noise identification task, designing a gating classifier based on the fusion of handcrafted features and semantic vectors.

C. Limitations of Existing Methods

Existing text denoising methods have the following limitations:

Blind spots in block noise identification. Pure rule-based methods rely on explicit features such as HTML tags and fixed keywords, and are almost powerless against block noise with complete surface structures and standardized grammar (such as navigation blocks, header/footer blocks, semantically incomplete policy fragments); pure semantic methods (such as LSA topic models) can capture semantic deviation but have limited ability to distinguish boundary samples with fuzzy semantic boundaries.

Pure deep learning methods have large parameter counts and high inference costs, making them unsuitable for enterprise local deployment.

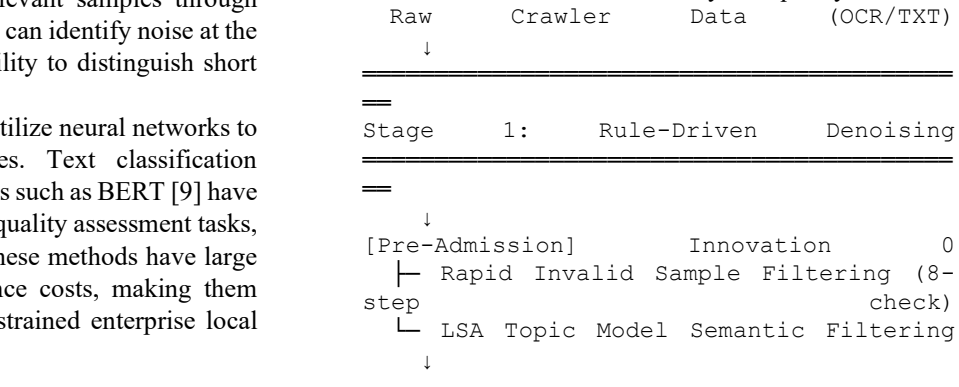
Lack of differentiated processing strategies for "obvious noise" versus "block noise."

The noise identification task lacks a large-scale labeled dataset and systematic evaluation benchmark oriented toward block noise.

III. Method

A. Overall Denoising Architecture

The two-stage denoising architecture proposed in this paper is shown in Figure 1. The core design philosophy is "rules first to filter obvious noise, intelligent gating to identify block noise": the first stage uses rule-based methods to quickly filter out obviously invalid samples, reducing the burden on the second stage; the second stage introduces a lightweight gating classifier specifically designed to identify block noise that rules cannot cover. Layered processing achieves a balance between efficiency and quality.



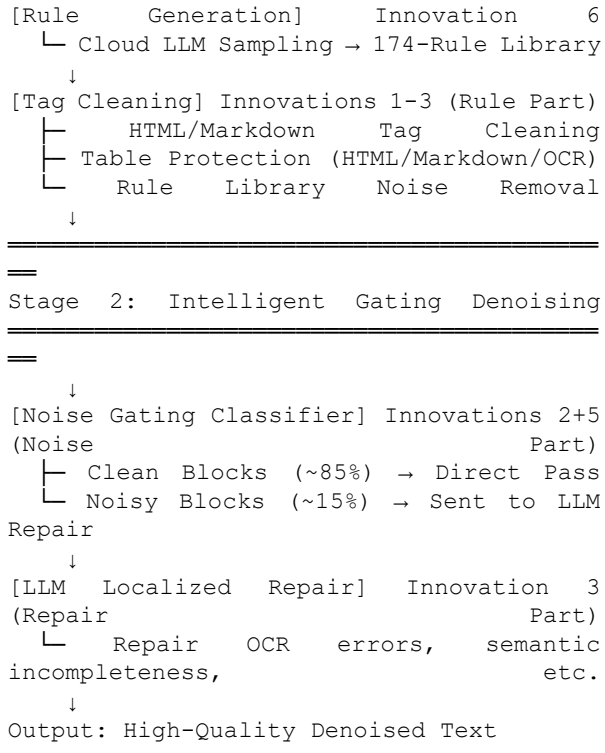


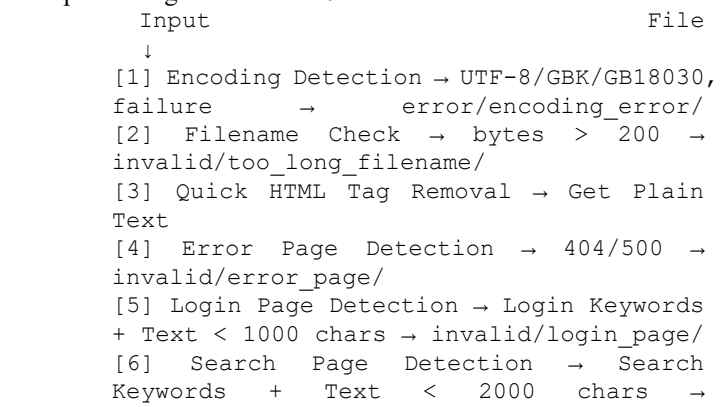
Fig. 1. Two-Stage Progressive Denoising Architecture

B. Pre-Admission Module

The pre-admission module is the first checkpoint in the pipeline, responsible for filtering obviously invalid files at the source, avoiding subsequent modules wasting computational resources.

1) Rapid invalid sample filtering

An 8-step check process is designed, with single file processing time below 10ms:



From the actual running data of 3500 samples from historical projects, this module achieves a filtering rate of 59.2% (valid 40.8%, invalid 59.2%), effectively reducing subsequent processing volume.

2) LSA semantic filtering

After invalid sample filtering, the LSA (Latent Semantic Analysis) topic model is introduced for secondary quality checking, identifying and filtering semantically irrelevant samples. Core technologies include:

Three-Tier Stopword Mechanism. To prevent LSA topic features from being overwhelmed by common function words, a three-layer stopword strategy is designed: the first tier consists of basic function words (approximately 40, such as "的", "了", "在"); the second tier consists of official document structural vocabulary (approximately 60, such as "关于", "通知", "印发"); the third tier consists of dynamic stopwords (automatically extracted top-N high-frequency words from the current dataset word frequency statistics, excluding business core word whitelists). Historical project validation shows that three-tier stopwords improve LSA topic discrimination by approximately 15% compared to a single stopword list.

Large-Sample-First Training Strategy. Policy original texts typically have more characters, while 404 pages or fragmented notifications have fewer characters. When training the LSA model, samples are sorted in descending order by file size, and the top 60% of the largest samples are selected as training corpus to ensure that the topic model learns the main semantic distribution from high-quality long documents.

Outlier Score Algorithm. A comprehensive outlier score is calculated for each document:

$$score = magnitude \times cosine\_similarity$$

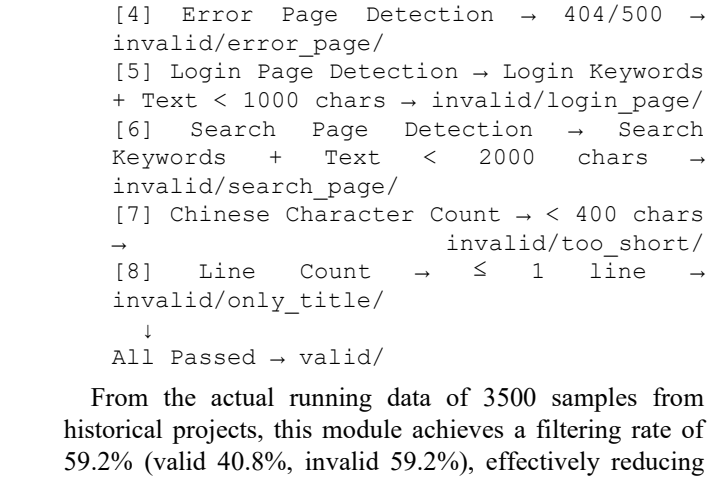
where magnitude measures the expression intensity of a document across all topics, and cosine similarity measures the directional consistency between the document and the average document vector. The lower the score, the more likely the document is an outlier.

A three-tier classification threshold is designed based on the score: score < 2.8 as definitely invalid, 2.8-5.0 as grey samples (pending review), >= 5.0 as valid samples.

C. Cloud LLM Rule Generation

Traditional rule library construction relies on manual writing, with high maintenance costs and limited coverage. This paper proposes a method of automatically generating noise identification rules based on cloud-based large model sampling.

Rule Generation Process:



The cost of this method is extremely low (approximately 1.5 RMB per 100 samples), and the generated rules, after frequency counting and deduplication, form a structured JSON rule library. The rule library can be directly called by the downstream tag cleaning module, achieving data-driven rule automation.

D. Dual-Mode Preprocessing Pipeline

For different application scenarios, two one-click switchable processing modes are designed:

Ultra-Fast Pure Mode (default): Uses a two-tier classification gating (normal/noisy), fast rule processing, zero modification to main text, prioritizing precision and processing efficiency. Applicable to scenarios with extremely high requirements for original text integrity where no modifications are allowed.

Enhanced Repair Mode (RAG scenario): Uses a three-tier classification gating (normal/noisy/pending repair), executing noise filtering and main text repair simultaneously, prioritizing semantic integrity. Applicable to RAG knowledge base construction scenarios that require maximizing information preservation.

The two modes share the same set of underlying denoising components, switching only the top-level strategy through configuration parameters, achieving flexible scenario-adaptive deployment.

E. Noise Gating Classifier

The noise gating classifier is the core component of the second stage, responsible for identifying complex noise that rule-based methods cannot cover. This paper proposes a hybrid classification strategy that fuses handcrafted statistical features with semantic centroid similarity.

1) Problem definition

Noise identification is formalized as a binary classification problem: given a text block, determine whether it is a "clean block" or a "noisy block."

Clean Block Criteria: No OCR recognition errors, no garbled characters, semantically complete (sentences are complete), clearly formatted, contains valid policy content.

Noisy Block Criteria: Contains OCR recognition errors (similar-shaped characters, character fragmentation), garbled characters (special characters, meaningless symbols), semantically incomplete (sentences are cut off), format corruption (HTML tag residues, mixed formats), contains irrelevant content (header/footer residues).

2) 32-Dimensional handcrafted statistical features

Based on the format and structural information of text blocks, 32-dimensional handcrafted statistical features are designed, as listed in Table 1:

Table 1. 32-Dimensional Handcrafted Statistical Features

Feature Dimension	Feature Name	Description
1	Line Count	Total number of lines in the text block
2	Total Character Count	Total number of characters in the text block
3	Average Line Length	Total characters / number of lines
4	Maximum Line Length	Character count of the longest line
5	Chinese Character Ratio	Chinese character count / total characters
6	Non-Chinese Character Ratio	Non-Chinese character count / total characters
7	Empty Line Ratio	Empty line count / total lines
8	HTML Tag Hit Count	Number of HTML tag occurrences in the text
9	Noise Keyword Hit Count	Hit count of predefined noise keywords (such as "advertisement", "promotion", etc.)
10	Digit Ratio	Digit character count / total characters
11	Punctuation Density	Punctuation mark count / total characters
12	Special Character Ratio	Special character count / total characters
13-32	(Other Format Statistics Features)	Indentation level, line length variance, consecutive empty line count, URL count, etc.

Feature Design Principle: All features are based on the format and statistical information of text blocks, without relying on any pre-trained models, ensuring extremely fast inference speed (< 1ms per block).

3) Noise semantic centroid construction

To introduce semantic-level discrimination capability, this paper proposes a noise representation method based on semantic vector centroids.

Semantic Encoding: The paraphrase-multilingual-MiniLM-L12-v2 model [13] (384 dimensions) is used to extract the semantic vector of each text block. This model has only approximately 22MB of parameters and can perform rapid inference on CPU.

Centroid Construction: Arithmetic mean is performed on the semantic vectors of training set noisy blocks, followed by L2 normalization, to obtain a single noise centroid. Further, K-Means clustering is used to construct a multi-centroid representation (K=3, based on 1706 noisy blocks in the training set). To prevent data leakage, centroid construction is strictly confined to the training set, and test set samples are only used for similarity calculation and evaluation.

Qualitative Validation: The constructed centroids are validated through nearest neighbor retrieval. Among the Top-5 nearest neighbors of the noise centroid, 4/5 are true noisy blocks; among the Top-5 nearest neighbors of the clean centroid, 3/5 are true clean blocks; the Top-50 overlap is only 8%, validating the discrimination between the two types of centroids in semantic space.

4) Feature fusion and classification model

Fusion Strategy: The maximum cosine similarity between the text block and the noise semantic centroid, and the maximum cosine similarity between the text block and the clean semantic centroid (both after normalization) are used as 2-dimensional semantic features, concatenated with 32-dimensional handcrafted features to form a 34-dimensional fused feature vector. Among them, clean centroid similarity serves as a reverse signal—the more similar to the clean centroid, the more likely it is a clean block.

Classification Model: A lightweight MLP (Multi-Layer Perceptron) architecture is adopted. Taking the dual-centroid fusion scheme as an example:

```
    Input: 34-dimensional Feature Vector
(32-D Handcrafted + 2-D Dual-Centroid
Similarity)
  ↓
Hidden Layer 1: 64-D, ReLU Activation
  ↓
```

```
Hidden Layer 2: 32-D, ReLU Activation
  ↓
Output Layer: 1-D, Sigmoid Activation
  ↓
Output: Clean Probability (0~1)
```

Inspired by model distillation and lightweight design principles [14], this paper adopts an extremely simple MLP architecture, with only approximately 4353 parameters, a size of about 22KB, and inference speed <1 ms per sample, easily deployable in enterprise local environments.

Training Configuration: Binary cross-entropy loss is adopted, Adam optimizer (lr=0.001), trained for 80 epochs, with ReduceLROnPlateau learning rate scheduling. Data split uses 80%/20% stratified sampling (random_state=42).

5) Threshold decision strategy

After the classifier outputs the clean probability, samples are classified into two categories through a threshold:

Threshold = 0.50 (default): Pursues high precision, suitable for scenarios sensitive to false positives.

Threshold = 0.25 (recall-optimized): Pursues high recall, suitable for scenarios sensitive to missed detections.

The threshold can be flexibly adjusted according to actual business needs, achieving a trade-off between precision and recall.

IV. Experiments

A. Dataset and Experimental Setup

1) Dataset construction

This paper constructs a noise identification dataset for web-crawled policy texts, containing a total of 4448 text blocks, of which 2315 are clean blocks and 2133 are noisy blocks. The dataset annotation criteria are as follows:

Clean Blocks: No OCR recognition errors, no garbled characters, semantically complete (sentences complete), clearly formatted, contains valid policy content.

Noisy Blocks: Contains OCR recognition errors, garbled characters, semantic incompleteness, format corruption, or irrelevant content residues.

The data comes from new energy vehicle policy texts crawled from government public websites, with manual block-by-block annotation after OCR recognition. The dataset is split using 80%/20% stratified sampling (random_state=42) to ensure consistent clean/noisy ratios between training and test sets, as shown in Table 2.

Table 2. Dataset Statistics

Dataset	Total Samples	Clean Blocks	Noisy Blocks	Ratio
Training Set	3558	1852	1706	52.1%/47.9%
Test Set	890	463	427	52.0%/48.0%
Total	4448	2315	2133	52.0%/48.0%

2) Experimental environment

All experiments were conducted in the following hardware environment, as shown in Table 3:

Table 3. Experimental Environment Configuration

Item	Configuration
CPU	Intel Core i7-12700
Memory	32GB DDR4
Operating System	Windows 10 / Ubuntu 20.04
Python	3.10
Deep Learning Framework	PyTorch 2.0

B. Evaluation Metrics

This paper adopts standard evaluation metrics for binary classification tasks:

Precision (P): the proportion of truly noisy samples among those predicted as noisy.

$$P = \frac{TP}{TP + FP}$$

Recall (R): the proportion of true noisy samples correctly identified.

$$R = \frac{TP}{TP + FN}$$

F1 Score: the harmonic mean of precision and recall.

$$F_1 = 2 \times \frac{P \times R}{P + R}$$

Where the positive class is defined as "noisy block." Since both missed detections (misjudging noise as clean) and false positives (misjudging clean as noise) have actual costs in the noise identification task, this paper uses the F1 score as the primary evaluation metric while also focusing on the balance between precision and recall.

C. Baseline Experiments

1) 32-dimensional handcrafted feature LightMLP baseline

First, the effectiveness of the 32-dimensional handcrafted statistical features (see Section 3.5.2) combined with a lightweight MLP classifier is validated. On the dataset of 4448 samples, using 80%/20% stratified sampling, the model architecture is 32→64→32→1, with approximately 4353 parameters, trained for 80 epochs. The results are presented in Table 4.

Table 4. 32-Dimensional Handcrafted Feature Baseline Results

Model	Feature Dimension	Threshold	Precision (%)	Recall (%)	F1(%)
LightMLP	32	0.35	75.74	72.37	74.01

The results show that using only format statistical features achieves an F1 score of 74.01% on 4448 samples, with precision 75.74% and recall 72.37%. This indicates that as sample size increases and noise types become more diverse, the expressive capacity of format features alone tends to saturate. This baseline provides a reference for subsequent semantic enhancement experiments.

D. Semantic Enhancement Experiments

1) Semantic centroid construction and qualitative validation

To introduce semantic-level discrimination capability, this paper first constructs noise semantic centroids. The paraphrase-multilingual-MiniLM-L12-v2 model (384 dimensions, approximately 22MB) is used to extract semantic vectors. Arithmetic mean is performed on training set noisy block semantic vectors followed by L2 normalization to obtain a single noise centroid; further, K-Means clustering is used to construct a multi-centroid representation (K=3, based on 1706 noisy blocks in the training set). To prevent data leakage, centroid construction is strictly confined to the training set.

Qualitative validation results: among the Top-5 nearest neighbors of the noise centroid, 4/5 are true noisy blocks; among the Top-5 nearest neighbors of the clean centroid, 3/5

are true clean blocks; the Top-50 overlap is only 8%, validating that the two types of centroids have discrimination in semantic space.

2) Pure semantic similarity classification

Cosine similarity between text blocks and noise centroids is used as a single semantic feature, evaluating pure semantic classification capability on 890 test set samples. The results are listed in Table 5.

Table 5. Pure semantic similarity results

Method	Centroid Count	Precision (%)	Recall (%)	F1(%)
Single Centroid	1	47.98	100.00	64.84
Multi-Centroid (K=3)	3	48.99	96.96	65.09
Multi-Centroid (K=4)	4	47.98	100.00	64.84
Multi-Centroid (K=5)	5	47.98	100.00	64.84

The F1 score of the pure semantic similarity method is approximately 65%, significantly lower than the 32-dimensional handcrafted feature baseline (74.01%). Analysis reveals: the semantic boundary between clean blocks and noisy blocks in policy texts is fuzzy, and some noisy blocks (such as OCR-erroneous policy clauses) highly overlap with clean blocks in semantic space; simultaneously, the similarity distribution is concentrated in the high-value interval [0.54, 0.96], with severe overlap between clean and noisy sample similarities. Although the pure semantic method performs limited independently (F1≈65%), its extremely high recall rate (96-100%) precisely complements the high precision (75.74%) of handcrafted features, providing a clear design direction for subsequent fusion.

E. Dynamic Semantic Clustering Experiments (Phase 3A)

To fully exploit the potential of semantic information, this paper further explores dynamic semantic clustering and dual-centroid strategies, validating the effectiveness of the semantic enhancement route under strict prevention of data leakage.

1) Dynamic-K Semantic Clustering

Traditional fixed-K multi-centroid representation may not cover the diversity of noise semantics. This paper proposes a dynamic-K strategy: pre-computing multiple groups of centroids for K=2~50 on training set noisy blocks, and automatically selecting the K value that maximizes similarity for each sample as the feature. The results are shown in Table 6.

Table 6. Dynamic-K semantic clustering results

Scheme	Description	Precision (%)	Recall (%)	F1(%)
Scheme A	Pure 32-D MLP	75.74	72.37	74.01
Scheme B	**Dynamic-K=[2,50] Ensemble**	**67.22**	**85.95**	**75.44**

Dynamic-K ensemble (Scheme B) achieves F1=75.44%, higher than the pure 32-D baseline (+1.43pp), with recall significantly improved to 85.95% (+13.58pp), but precision drops to 67.22%. The most frequently selected K values are K=10 (33.8%) and K=9 (26.7%), indicating high semantic heterogeneity of noise texts, requiring many fine-grained clusters for coverage.

2) Dual-Centroid Semantic Classifier

In addition to the noise centroid, this paper further constructs a clean centroid, forming a dual-centroid signal: the higher the similarity to the noise centroid, the more likely it is noise; the lower the similarity to the clean centroid, the more likely it is noise. To prevent data leakage, KMeans uses only training set noisy/clean samples respectively to construct centroids.

Test set feature distribution shows: clean centroid discrimination (clean mean 0.8333 vs noisy mean 0.7094, difference 0.1239) is much greater than noise centroid (difference 0.0275), indicating that clean semantic space is more concentrated and the reverse signal is stronger. The detailed results are shown in Table 7.

Table 7. Dual-centroid ensemble results

Scheme	Composition	Precision (%)	Recall (%)	F1(%)
Scheme A	Pure 32-D MLP	75.74	72.37	74.01
Scheme B	Dynamic-K=[2,50] Ensemble	67.22	85.95	75.44
Scheme C	**Dual-Centroid Ensemble**	**72.22**	**82.20**	**76.89**
Scheme C_v2	Dual-Centroid MLP Fusion	74.11	82.44	**78.05**

Dual-centroid Scheme C (F1=76.89%) outperforms dynamic-K Scheme B (75.44%), with significant precision improvement (+4.78pp). Further feeding dual-centroid similarity into SmallMLP(2→16→8→1) for independent weight learning (Scheme C_v2), F1 improves to 78.05% (compared to 32-D baseline +4.04pp), which is the best result for the dynamic semantic clustering route.

F. Experimental Results Summary and Comparison

Table 8. Summary of Qs (4448 Samples, Test Set 890)

No.	Method	Feature Dimension	Precision (%)	Recall (%)	F1(%)	Parameters	Model Size
1	32-D Handcrafted LightMLP	32	75.74	72.37	74.01	~4353	~22KB
2	Single Centroid Similarity	1	47.98	100.00	64.84	-	-
3	Multi-Centroid K=3 Similarity	1	48.99	96.96	65.09	-	-
4	Dynamic-K=[2,50] Ensemble	1+32	67.22	85.95	75.44	~4450	~22KB
5	**Dual-Centroid MLP Fusion (C_v2)**	2+32	**74.11**	**82.44**	**78.05**	~4480	~23KB

Experimental results indicate (see Table 8):

Handcrafted Feature Baseline: 32-dimensional handcrafted features achieve 74.01% F1 on 4448 samples, providing a baseline for subsequent enhancement.

Pure Semantic Methods Are Limited: Single/multi-centroid similarity F1 is only approximately 65%, indicating fuzzy clean/noisy semantic boundaries in policy texts; semantic signals need to be fused with format features to be effective.

Dual-Centroid MLP Fusion Is Optimal: Dual-centroid MLP fusion achieves 78.05% F1 (precision 74.11%, recall 82.44%), which is the best result under our experimental

conditions. This scheme achieves a +4.04pp improvement over the 32-D baseline with only an approximately 4480-parameter, approximately 23KB extremely lightweight architecture, and recall is significantly higher than the baseline (+10.07pp), validating the effectiveness of dual-centroid semantic signals.

G. End-to-End Analysis

1) Pre-filtering efficiency

The pre-admission module processed 3500 samples in actual projects, with results shown in Table 9:

Table 9. Pre-admission module filtering effect

Filtering Stage	Samples Processed	Valid Samples	Invalid Samples	Filtering Rate
Rapid Invalid Sample Filtering	3500	1428 (40.8%)	2072 (59.2%)	59.2%
LSA Semantic Filtering	1428	Approximately 1200	Approximately 228	Approximately 16%

>Note: LSA semantic filtering data is approximate statistics based on historical project running logs; some samples have fuzzy boundaries and were not precisely annotated one by one.

Rapid invalid sample filtering has a single file processing time < 10ms, with a total processing time of approximately 35 seconds on 3500 samples, filtering out 59.2% of obviously invalid files (encoding errors, error pages, login pages, too-short texts, etc.), greatly reducing the computational burden on subsequent modules.

2) Overall pipeline efficiency

The end-to-end processing efficiency of the two-stage pipeline is summarized in Table 10.

Table 10. Two-stage denoising pipeline efficiency analysis

Stage	Module	Processing Speed	Coverage	Description
Stage 1	Rapid Filtering	<10ms/file	~59%	Filters obviously invalid samples
Stage 1	LSA Semantic Filtering	~50ms/file	~16%	Filters semantically irrelevant samples
Stage 1	Rule Cleaning	~5ms/block	~90%	Cleans HTML tags, advertisements, etc.
Stage 2	Gating Classifier	<1ms/block	~15%	Identifies block noise
Stage 2	LLM Repair	~500ms/block	<5%	Processes only difficult samples

The overall pipeline design follows the "low-cost first, high-precision fallback" principle: approximately 75% of samples complete denoising by rule-based methods within millisecond-level latency in Stage 1; approximately 15% of samples enter the Stage 2 gating classifier; only less than 5% of the most difficult samples require LLM repair. This layered processing strategy keeps the average processing cost at an extremely low level while ensuring denoising quality.

V. Conclusion and Future Work

A. Summary of Work

This paper addresses the block noise identification challenge for web-crawled policy texts in enterprise RAG systems, proposing a two-stage progressive denoising method, with the following main contributions:

(1) A "rule-first, intelligent fallback" two-stage denoising architecture oriented toward block noise identification is designed. The first stage quickly filters out obviously invalid samples through rapid invalid sample filtering, LSA topic model semantic filtering, and a 174-rule library generated by cloud LLM sampling, reducing the burden on the second stage; the second stage introduces a lightweight gating classifier specifically designed to identify block noise that rules cannot cover.

(2) A block noise gating classification strategy fusing 32-dimensional handcrafted statistical features with dual-centroid semantic similarity is proposed. To address the core challenge of high surface feature similarity between block noise and clean text, handcrafted features capture subtle format-level differences, while dual-centroid semantic similarity introduces reverse signals from noise/clean semantic spaces, ultimately training a lightweight MLP classifier with only approximately 4480 parameters. Experimental results show that the dual-centroid MLP fused classifier achieves 78.05% F1 on 4448 samples (precision 74.11%, recall 82.44%), significantly outperforming the 32-dimensional handcrafted feature baseline (+4.04pp), validating the effective identification capability of this strategy for block noise.

(3) A complete engineering denoising pipeline is constructed, including pre-admission, rule generation, tag cleaning, gating classification, and other modules, all of whose core components can be locally deployed, meeting enterprise privacy security requirements.

B. Main Innovations

The main innovations of this paper include:

Block Noise Gating Classifier: Handcrafted Features Fused with Dual-Centroid Semantics. To address the core challenge of high surface feature similarity between block noise and clean text, a hybrid classification strategy fusing 32-dimensional format statistics features with dual-centroid semantic similarity is designed: handcrafted features capture subtle format-level differences of block noise, and dual-centroid semantic similarity introduces reverse signals from noise/clean semantic spaces. A single MLP has only approximately 4353 parameters and approximately 22KB size; dual-centroid MLP fusion achieves 78.05% F1 (74.11%/82.44%) with an extremely lightweight architecture of approximately 4480 parameters and approximately 23KB, which is the current best result.

Two-Stage Progressive Architecture for Block Noise Identification. A two-stage architecture is designed: "rules first to filter obvious noise, intelligent gating to identify block noise." The first stage quickly filters obvious noise through rapid invalid sample filtering, LSA topic model, and rule library, reducing the burden on the second stage; the second stage gating classifier specifically handles block noise that rules cannot identify, achieving "low-cost first,

high-precision fallback."

Cloud LLM Sampling Rule Generation. An automatic rule generation method based on the DeepSeek API is proposed, generating 174 structured noise identification rules at extremely low cost (approximately 1.5 RMB per 100 samples), achieving data-driven automated rule library construction, filtering obvious pattern noise for the gating classifier.

C. Limitations and Future Work

This paper's work still has the following limitations, worthy of future research:

(1) **Limited Dataset** Scale. The current dataset contains only 4448 text blocks, with noise types mainly concentrated on common types such as OCR errors, HTML residues, and semantic incompleteness. Future work can expand the annotation scale to cover more noise types (such as multilingual mixed text, complex table corruption, etc.).

(2) **Domain Transferability of Semantic Centroids**. The current noise semantic centroids are constructed based on new energy policy texts, and their applicability in other domains (such as medical, legal texts) remains to be validated. Future work can explore domain-adaptive centroid update mechanisms.

(3) **Cost and Latency of LLM Repair**. Although this paper controls the LLM invocation ratio to below 5% through the two-stage architecture, single LLM repair still requires approximately 500ms. Future work can research lighter-weight repair models (such as locally deployed small models fine-tuned based on LoRA) to further reduce latency and cost.

(4) **Multi-Task Joint Optimization**. The denoising task in this paper is independently optimized from downstream RAG retrieval and generation tasks, without considering the impact of denoising strategies on retrieval accuracy. Future work can explore denoising-retrieval joint optimization frameworks, guiding denoising strategies with the goal of end-to-end retrieval quality improvement.

CONFLICT OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459-9474, 2020.

[2] Y. Gao *et al.*, "Retrieval-augmented generation for large language models: A survey," arXiv preprint arXiv:2312.10997, 2023.

[3] S. S. Bhamare and B. V. Pawar, "Survey on web page noise cleaning for web mining," *International Journal of Computer Science and Information Technologies*, vol. 4, (5), pp. 662-665, 2013.

[4] J. Bevendorff *et al.*, "An empirical comparison of web content extraction algorithms," in *Proc. 46th International ACM SIGIR Conf; on Research and Development in Information Retrieval*, pp. 1488-1497, 2023.

[5] W. X. Zhao *et al.*, "A survey of large language models," arXiv preprint arXiv:2303.18223, 2023.

[6] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," in *Proc. Third ACM International Conf. on Web Search and Data Mining*, 2010, pp. 441-450.

[7] T. Weninger, W. H. Hsu, and J. Han, "CETR: Content extraction via tag ratios," in *Proc. 19th International Conf. on World Wide Web*, 2010, pp. 971-980.

[8] S. Deerwester *et al.*, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, (6), pp. 391-407, 1990.

[9] J. Devlin *et al.*, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171-4186.

[10] L. V. Subramaniam *et al.*, "A survey of types of text noise and techniques to handle noisy text," in *Proc. Workshop on Analytics for Noisy Unstructured Text Data*, 2009v115-122.

[11] S. Minaee *et al.*, "Deep learning-based text classification: A comprehensive review," *ACM Computing Surveys*, 54(3), pp. 1-40, 2021.

[12] C. Xu and J. McAuley, "A survey on dynamic neural networks for natural language processing," *Findings of the Association for Computational Linguistics: EACL 2023*, 2023, pp. 2200-2217.

[13] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. 2019 Conf. on Empirical Methods in Natural Language Processing*, 2019, pp. 3982-3992.

[14] X. Jiao *et al.*, "TinyBERT: Distilling BERT for natural language understanding," *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 4163-4174.