Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 8, No. 2, 2026

A Statistical Comparison of Multi-Architecture Models for Retention-Period Classification of Chinese Electronic Official Documents

Haoran Zhang [1], Lili Shi [1], Tuochen Pan [2]

[1] School of Artificial Intelligence, Wenzhou Polytechnic University, Wenzhou, Zhejiang 325035, China
[2] Ruian College, Wenzhou Polytechnic University, Wenzhou, Zhejiang 325200, China

Abstract: Electronic document archiving requires automatic retention-period classification (permanent/30-year/10-year). To evaluate architectures on weakly structured Chinese official texts—texts lacking clear semantic boundaries whose key cues appear as scattered keywords—we benchmark six models across four paradigms: Qwen3 (0.6B/1.7B/4B) with LoRA, BERT-base-Chinese, FastText, and BERT-Embedding-TextCNN (frozen BERT character embeddings + TextCNN). On 2,118 test samples we conduct pairwise McNemar exact binomial tests with Benjamini-Hochberg correction over the 15 comparisons (FDR q=0.05). BERT-Embedding-TextCNN (75.50%) and FastText (75.07%) achieve the highest accuracy, ahead of Qwen3-4B (73.94%), but their Wilson 95% confidence intervals (approximately ±1.9 percentage points) overlap substantially. After correction, only two differences remain significant: BERT-Embedding-TextCNN and FastText each outperform Qwen3-0.6B (adjusted p=0.002 and p=0.006); the other 13 pairwise differences do not reach significance. The accuracies thus form descriptive clusters rather than statistically verified tiers. A leakage-controlled evaluation on the 1,690 test documents that are neither exact nor near-duplicates of training documents preserves this ranking and both significant differences, with accuracies 1.7–2.5 percentage points lower across models. Feature-weight inspection shows that archival keywords (e.g., bidding and state-asset terms, which skew toward permanent retention) are important predictive features for FastText, supporting a keyword-driven explanation of task success. For resource-constrained enterprise deployment, FastText offers near-top accuracy with CPU-only inference at roughly 26,800× the throughput of the 4B LLM under the evaluated hardware configurations.

Keywords: Electronic Document Archiving, Text Classification, McNemar Test, Benjamini-Hochberg Correction, Large Language Models, FastText, Statistical Significance.

1. Introduction

With the deepening of digital transformation, enterprise business models are accelerating the shift from primarily manual, offline operations to digitally-driven online models [1]. This process has led to increasing digitization of internal enterprise activities, generating massive volumes of electronic documents with preservation value and legal validity. The systematic classification and long-term preservation of such electronic documents are referred to as electronic document archiving. Its core objective is to leverage modern information technology to transform electronic data into digitally managed archival resources, ensuring the authenticity, integrity, usability, and security of documents [2].

However, traditional electronic document archiving management primarily relies on manual operations for classification, labeling, and storage. This approach is not only labor-intensive and time-consuming, but also struggles to ensure efficient, accurate archiving and convenient subsequent retrieval when processing massive, heterogeneous electronic documents [3]. Among these processes, the determination of retention periods is the most experience-dependent step—practitioners must comprehensively parse the document's red-header structure, responsible unit, and dispatch intent, then combine these with the body content to identify the archival category (e.g., distinguishing "personnel appointments" from "disciplinary actions"), and finally determine the retention period as permanent, 30-year, or 10-year in accordance with the Archives Law and industry regulations. This process involves extensive professional rules and semantic understanding, making manual operations prone to misclassification and omission.

In recent years, artificial intelligence technologies based on the Transformer architecture [4], exemplified by Large Language Models (LLMs), have developed rapidly, propelling society into the "AI Plus" era. In 2021, the General Office of the CPC Central Committee and the General Office of the State Council issued the 14th Five-Year Plan for National Archives Development [5], explicitly stating the need to "strengthen the application of big data, artificial intelligence, and other new-generation information technologies in the construction of digital archives." However, when deploying AI technologies in archival management scenarios, a fundamental question remains insufficiently answered: which model architecture is most suitable for the retention-period classification of electronic official documents?

Currently, multiple architectural paradigms exist in the text classification domain, ranging from traditional n-gram shallow models (e.g., FastText [6]) to deep pre-trained language models (e.g., BERT [7]), and further to instruction-tuned large language models (e.g., the Qwen3 series [8]). These architectures differ enormously in parameter scale, inference speed, deployment cost, and classification performance. In resource-constrained enterprise deployment environments (e.g., CPU-only servers, low-latency requirements), selecting the appropriate model architecture becomes an engineering problem requiring empirical evidence. More critically, existing studies mostly rely on

simple comparisons of accuracy figures, lacking rigorous testing of whether inter-model differences are statistically significant, potentially leading to over-interpretation of minor performance differences.

This study addresses the above issues by systematically comparing the performance of six model architectures on the three-class classification of electronic official document retention periods, and conducting McNemar exact binomial tests [9] for statistical significance analysis of inter-model differences. Specifically, the study focuses on three core questions. First, on weakly structured Chinese official document texts, are the accuracy differences among models of different architectural paradigms (shallow n-gram, convolutional neural networks, pre-trained language models, large language models) statistically significant? Second, does increasing model parameter scale consistently improve classification performance across the evaluated model configurations? Do large language models necessarily outperform lightweight models? Third, for enterprise electronic document archiving deployment, which model architecture achieves the optimal balance between accuracy and inference efficiency?

2. Literature Review

2.1. Current Research on Intelligent Electronic Document Archiving

Research on intelligent electronic document archiving began relatively early internationally. The Proof of Concept (PoC) project in Victoria, Australia [10] employed eDiscovery and Nuix machine learning tools to assist in the validity appraisal of email archives; the AI for Selection project led by The National Archives of the UK [11] tested the application of machine learning techniques in electronic archive appraisal, screening, and classification review. In archival retrieval, the European EHRI project [12] utilized AI to automatically synchronize multi-source archival metadata; the EU "Time Machine" project [13] developed an image archive retrieval system based on convolutional neural networks.

In China, Sun Yat-sen University employed metadata technology to achieve automatic classification of electronic documents from business systems [14]; the Shanghai Archives' "multi-agent" system [15] decomposed complex systems into manageable subsystems, achieving intelligent management across archiving, management, and transfer phases. Chen Xushan [16] noted the natural close connection between archival work and AI, constructing a foundational application framework for AI in the archival domain. However, the above studies mostly focus on macro-level process design of archival management or the processing of image/audio-visual archives, with relatively few studies on fine-grained model comparison specifically for text-based electronic document retention-period classification. In particular, to the best of our knowledge, few studies have systematically combined multi-paradigm architectures (from shallow n-gram models to LLMs) with multiple-comparison-corrected significance testing on this task; this constitutes the research gap addressed by the present study.

2.2. Text Classification Model Architectures

Text classification is a fundamental task in natural language processing, having evolved from shallow to deep models. FastText [6], proposed by Facebook in 2016, is based on the bag-of-words model with n-gram features, achieving efficient text classification through a shallow neural network. With its extremely fast training speed and competitive accuracy, it has become a classic strong baseline model. TextCNN [17] utilizes convolutional neural networks to capture local n-gram features, achieving strong performance on multiple benchmark datasets.

In 2018, the introduction of BERT [7] marked the beginning of the pre-trained language model era. BERT employs a bidirectional Transformer encoder for self-supervised pre-training on large-scale corpora, followed by fine-tuning for downstream tasks, achieving breakthroughs on multiple NLP benchmarks. However, BERT's 12-layer self-attention mechanism incurs substantial computational overhead. In recent years, the channel attention mechanism proposed by SE-Net [18] has been introduced into text classification. Yuan and Zou [19] proposed SECNN, combining the SE module with CNN for Chinese sentence classification, enabling the model to learn the importance of convolutional features at different scales.

In combining pre-trained language models with convolutional networks, several studies have explored various architectures. Some studies feed BERT's contextual embeddings into CNNs, forming hybrid architectures such as BERT-CNN-SE [25], BERT-SCNN [26], and BERT-Att-TextCNN [27]. The shared idea behind these approaches is to leverage BERT's high-quality pre-trained representations while using CNNs' local receptive fields to reduce reliance on deep self-attention. However, these studies mainly target general text classification or dialogue recognition tasks; to the best of our knowledge, little prior work has systematically evaluated such hybrid architectures on weakly structured electronic document retention-period classification, nor has the necessity of the SE module been rigorously ablated. Drawing on SE-Net [18] and SECNN [19], this study constructs the BERT-Embedding-TextCNN model (experiment ID: bert_char) and explores, through ablation experiments, the effects of frozen BERT embeddings, the SE module, and pooling strategies on official document classification.

In the domain of large language models, generative pre-trained models represented by GPT [20], LLaMA [21], and Qwen [8] have demonstrated powerful zero-shot and few-shot capabilities through parameter scaling (billions to hundreds of billions) and instruction tuning. Open-source frameworks such as LLaMA-Factory [22] have lowered the engineering barrier for LLM fine-tuning. However, whether LLMs outperform traditional lightweight models on text classification tasks in specific vertical domains (e.g., archival management) remains an open question. Existing studies have shown that on certain tasks, fine-tuned small models can match or even exceed the performance of large models [23].

2.3. Statistical Significance Testing in Model Evaluation

In machine learning model comparison, relying solely on accuracy figures can be misleading, especially when differences are small (e.g., 1%–2%). Dietterich [24] emphasized the importance of statistical significance testing in model evaluation. The McNemar test [9] is a classic method for paired classification model comparison, based on constructing a 2×2 contingency table from per-sample predictions of two models on the same test set, and determining whether the error patterns of the two models

differ significantly through an exact binomial test. Compared to the t-test, the McNemar test does not require normality assumptions and is better suited for paired comparisons of classification models.

However, the application of statistical significance testing in existing text classification research remains insufficient. Many studies report only accuracy rankings without testing whether inter-model differences are statistically meaningful. Moreover, when multiple models are compared pairwise, the family of tests inflates the false-positive rate; false-discovery-rate procedures such as the Benjamini-Hochberg correction [29] should accompany any multiple comparison. This study applies the McNemar test to all 15 pairwise comparisons among six models and reports both raw and BH-adjusted p-values.

3. Current Status and Problem Analysis

This study is grounded in the electronic document archiving practice of the Wenzhou Railway and Rail Transit Investment Group Co., Ltd. Through in-depth investigationof the enterprise's existing archiving workflow, three core problems were identified.

First, official document texts are distinctly weakly structured. Unlike structured texts such as academic papers and news articles, electronic official documents often lack clear semantic boundaries between titles and body text. The determination of retention periods for official documents relies heavily on keyword features (e.g., "personnel appointment" corresponds to permanent, "general notice" corresponds to 10-year), rather than deep semantic reasoning. This characteristic may give shallow n-gram models a natural advantage on this task, while the complex reasoning capabilities of deep semantic models may become a "liability."

Second, there is a lack of empirical evidence for model selection. In the practice of deploying AI technologies in archival management, there is a tendency toward the "bigger is better" assumption, namely that models with larger parameter scales (e.g., LLMs) necessarily outperform lightweight models. However, whether this assumption holds on weakly structured text classification tasks lacks systematic verification. Enterprise deployment environments typically face constraints such as CPU inference and low latency (<0.1s), requiring a balance between accuracy and efficiency in model selection.

Third, the statistical significance of performance differences is ambiguous. Existing model comparison studies mostly report accuracy figures, but when the accuracy difference between two models is only 1%–2%, this difference may arise from sample randomness rather than fundamental differences in model capability. The lack of statistical significance testing may lead to over-interpretation of minor differences, thereby affecting the scientific validity of deployment decisions.

4. Research Objectives and Contents

As shown in Fig. 1, this study focuses on the three-class classification of electronic official document retention periods (permanent/30-year/10-year), systematically comparing six models across four architectural paradigms, and conducting statistical significance analysis through the McNemar test with Benjamini-Hochberg correction.

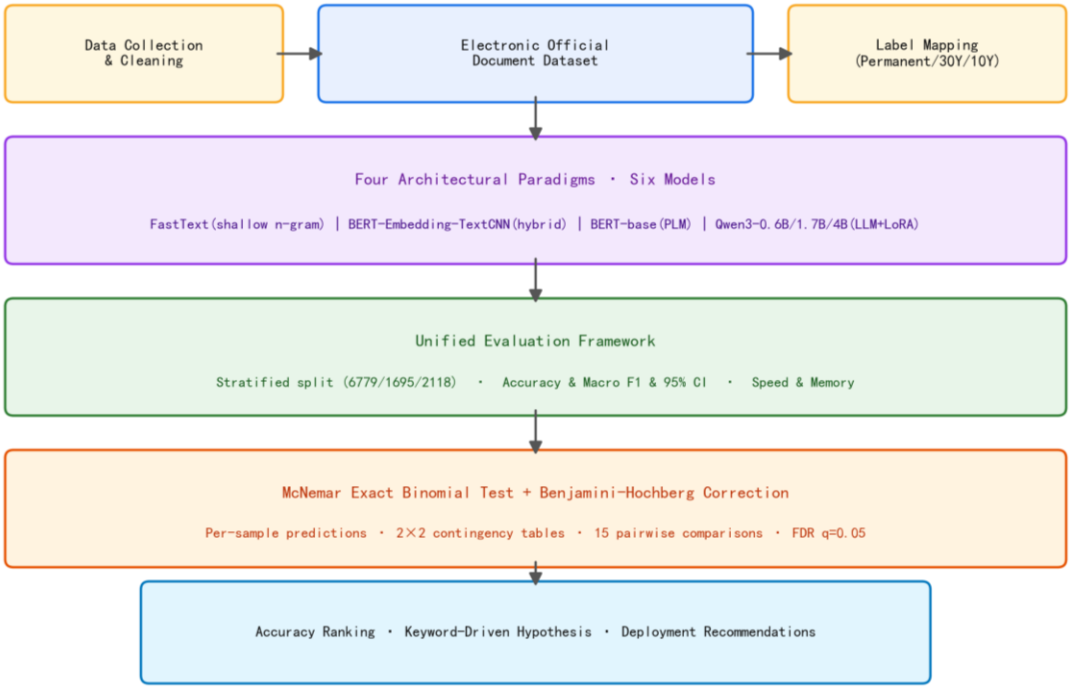

Figure 1. Research framework

The study comprises the following four components. First, multi-architecture model construction, covering four architectural paradigms from shallow to deep: shallow n-gram models (FastText), convolutional hybrid models (BERT-Embedding-TextCNN), pre-trained language models (BERT-base-Chinese), and large language models (Qwen3-0.6B/1.7B/4B + LoRA fine-tuning). Second, a unified evaluation framework, using the same stratified training/validation/test split (6,779/1,695/2,118 samples), with accuracy and Macro F1 (with Wilson 95% confidence intervals) as core metrics, and inference speed and memory usage recorded to ensure fairness in model comparison. Third, statistical significance testing, conducting McNemar exact binomial tests on per-sample predictions of the six models, constructing a p-value matrix of 15 pairwise comparisons, and applying Benjamini-Hochberg false-discovery-rate

correction. Fourth, extraction of key findings, analyzing the relationship between model architecture and task characteristics, examining the keyword-driven classification hypothesis through explicit feature-weight inspection, and providing deployment recommendations for resource-constrained scenarios.

5. Methodology

5.1. Dataset

The experimental data is sourced from the OA system of the Wenzhou Railway and Rail Transit Investment Group Co., Ltd., comprising administrative official documents from the past five years. After desensitization, cleaning, and business label mapping, a structured dataset was formed. Each sample consists of two text fields—the responsible unit and the document title (a median length of 44 characters)—together with the retention-period label determined in accordance with the Archives Law and the Regulations on the Administration of Enterprise Archives [28]. The dataset is divided into 6,779 training, 1,695 validation, and 2,118 test samples by stratified random sampling, preserving identical class proportions across the three splits (30-year 43.2%, 10-year 31.7%, permanent 25.1%), as shown in Table 1. For transparency, we disclose a data-quality issue: 106 test documents (5.0%) are exact duplicates of training documents in their full input text (responsible unit and title), and a further 322 (15.2%) have a character-bigram Jaccard similarity above 0.8 with at least one training document (20.2% of the test set in total). These overlaps may inflate the reported test performance and limit the extent to which the results can be interpreted as performance on previously unseen document formulations; Section 6.6 therefore reports a leakage-controlled evaluation on the remaining 1,690 test documents.

Table 1. Test set label distribution

Retention Period	Samples	Proportion	Determination Basis
Permanent	531	25.1%	Major decisions, personnel appointments
30-year	915	43.2%	Routine administrative affairs
10-year	672	31.7%	General transactional notices

5.2. Model Architectures

The six models compared in this study are categorized into four architectural paradigms, as shown in Table 2.

Table 2. Model architecture overview

Model	Paradigm	Parameters	Training Method	Key Configuration
FastText	Shallow n-gram	200.7M allocated (sparse hash; ≈6.8M effective)	Train from scratch	lr=0.3, epoch=50, ngram=3, dim=100
BERT-Embedding-TextCNN	Hybrid	16.68M total / 0.46M trainable	BERT frozen emb + CNN fine-tune	embed_dim=200, filters=100, kernels=(1, 2, 3, 4, 5)
BERT-base-Chinese	Pre-trained LM	102M	Full fine-tuning	lr=2e-5, batch=32, max_len=512
Qwen3-0.6B + LoRA	LLM	0.6B	LoRA fine-tuning (r=16)	FP16, batch=16, max_new_tokens=16
Qwen3-1.7B + LoRA	LLM	1.7B	LoRA fine-tuning (r=16)	FP16, batch=4
Qwen3-4B + LoRA	LLM	4B	LoRA fine-tuning (r=16)	8-bit quantization, batch=4

FastText employs a word n-gram bag-of-words representation, mapping to label space through a single hidden layer, serving as a strong baseline with extremely fast inference.

BERT-Embedding-TextCNN (referred to as bert_char in our experiment logs) is a lightweight hybrid architecture constructed in this study, drawing on the channel-attention-and-convolution fusion idea of SE-Net [18] and SECNN [19]. Importantly, the model does not execute the BERT Transformer encoder: it uses only the pre-trained character-embedding table of BERT-base-Chinese (21,128 entries × 768 dims, 16.22M parameters) as a frozen lookup, projects the 768-dim vectors to 200 dimensions through a linear layer, and feeds the result into a TextCNN with convolution kernels of sizes 1/2/3/4/5 (100 filters each) followed by global max pooling and a fully connected classifier. Only the projection, CNN, and classifier parameters (0.46M) are trained; total parameters are 16.68M including the frozen embedding table. The ablation study in Table 3 shows that the configuration without the SE module achieves the best accuracy (75.50%) on this task, while replacing the BERT embeddings with randomly initialized embeddings drops accuracy to 72.24%, indicating that the pre-trained character representations—not deeper contextual encoding—are the useful component.

The SE module computes channel-wise importance weights for convolutional features. Let z_c denote the globally pooled feature of channel c; the channel attention weight s_c is obtained through a two-layer fully connected gate:

$$S_c = \sigma(W_2 \cdot \mathrm{ReLU}(W_1 \cdot z_c))$$

Equation (1) SE channel attention weight

Where $\sigma(\cdot)$ is the sigmoid function and W_0 and W_1 are dimensionality-reduction/restoration matrices. In the ablation experiments of this study, removing this module improved accuracy, suggesting that explicit convolutional keyword extraction is already sufficient for weakly structured official documents.

Table 3. Ablation of BERT-Embedding-TextCNN components

Configuration	Trainable Params	Accuracy	Macro F1
BERT frozen embeddings + TextCNN (final, no SE)	0.46M	75.50%	75.66%
BERT frozen embeddings + TextCNN + SE attention	0.58M	74.79%	74.69%
BERT frozen embeddings + multi-pooling (max + avg)	0.46M	74.98%	74.85%
Random embeddings + TextCNN	0.96M	72.24%	72.06%

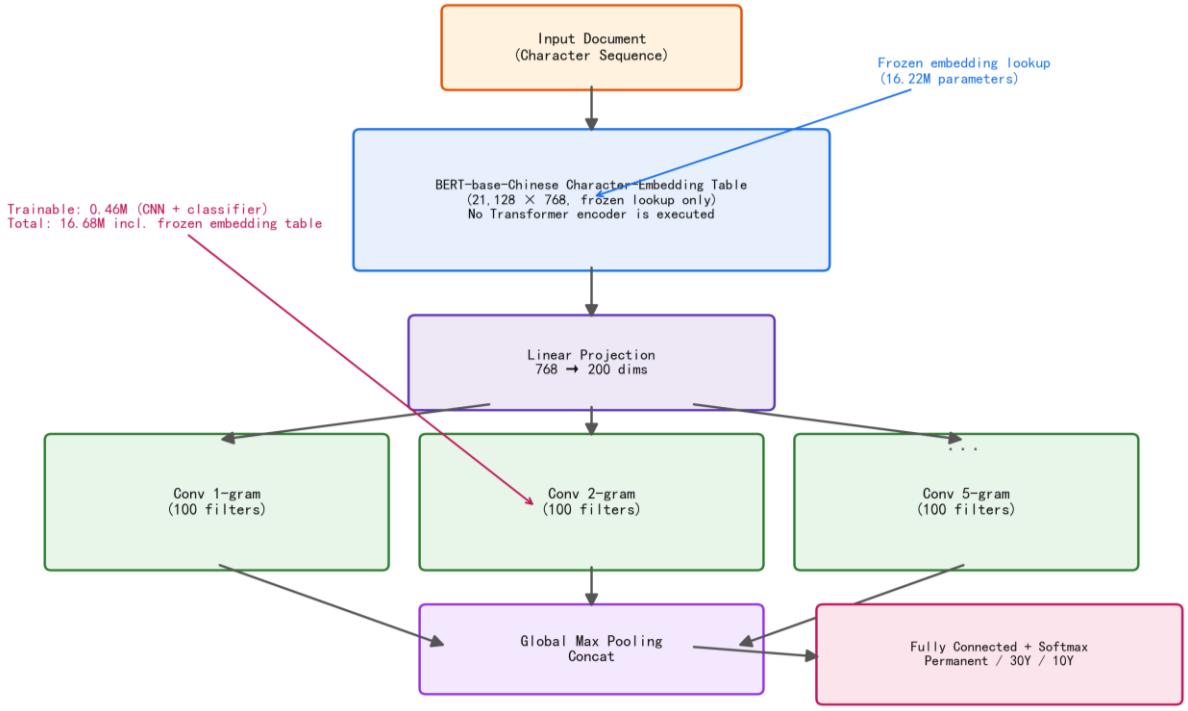

Figure 2. Architecture of BERT-Embedding-TextCNN: frozen character-embedding lookup (no Transformer encoder) + TextCNN

BERT-base-Chinese employs standard full-parameter fine-tuning, in which all 12 Transformer encoder layers are updated.

The Qwen3 series models are fine-tuned with LoRA (Low-Rank Adaptation), which attaches low-rank adapters (rank=16) to the Qwen3-Instruct base model; the Qwen Chat Template is used for both training and inference. Qwen3-4B uses 8-bit quantization to fit within the 8GB VRAM constraint.

LoRA introduces a low-rank increment $\Delta W = BA$ alongside the original weight matrix W_0, training only the low-rank matrices B and A to substantially reduce memory and computation costs. The forward pass can be written as:$\Delta W=BA\,W_0BA$

$$h=W_0x+\Delta Wx=W_0x+BAx$$

Equation (2) LoRA low-rank adaptation

Where $B\in Rd\times r$, $A\in Rr\times k$, $r\ll\min(d,k)$, r=16. r is the rank and $r \ll d$. This study sets r = 16, optimizing only the low-rank adapters while keeping the base parameters frozen.

5.3. Training and Evaluation

All models are trained on the same training set. Hyperparameters are selected and early stopping is applied on the validation set; final performance is reported on the test set. Evaluation metrics include Accuracy, Macro F1 with Wilson 95% confidence intervals, inference speed (samples/second), and VRAM usage (GB). Inference benchmarks were conducted on a single NVIDIA GeForce RTX 4060 Laptop GPU (8GB VRAM) with PyTorch 2.6.0 (CUDA 12.4) under Windows 11 (32GB RAM): FastText runs single-sample prediction on CPU; BERT-Embedding-TextCNN and BERT-base use batched GPU inference with batch size 64; Qwen3 models use batch size 4, max_new_tokens=16, and greedy decoding. Each measurement is preceded by warm-up passes and uses CUDA synchronization before and after each batch; LLM throughput includes tokenization and generation overhead.

Accuracy is the proportion of correctly predicted samples. Let N be the number of test samples, y_i the true label and $\hat{y}_i$ the predicted label for sample i, and $I(\cdot)$ the indicator function that equals 1 when the condition holds and 0 otherwise:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} I(\hat{y}_i = y_i)$$

Equation (3) Accuracy

For each class c, precision P_c, recall R_c, and the F1 score $F1_c$ are defined as:

$$P_c = \frac{TP_c}{TP_c + FP_c}$$

Equation (4) Class precision

$$R_c = \frac{TP_c}{TPc+FNc}$$

Equation (5) Class recall

$$F1_c = \frac{2\cdot Precision_c \cdot Recall_c}{Precision_c + Recall_c}$$

Equation (6) Class F1 scoreF1

Macro F1 is the arithmetic mean of the per-class F1 scores over all C classes:

$$\text{Macro F1} = \frac{1}{C}\sum_{c=1}^{C} F1_c$$

Equation (7) Macro F1

5.4. McNemar Significance Test

Per-sample predictions (correct/incorrect binary markers) of the six models are subjected to pairwise McNemar exact binomial tests. For each pair of models A and B, a 2×2 contingency table is constructed, where a denotes both models predicting correctly, b denotes A correct and B incorrect, c denotes A incorrect and B correct, and d denotes both models predicting incorrectly. Under the null hypothesis (no difference between the two models), among the discordant pairs (b+c), b and c follow a binomial distribution B(b+c, 0.5). The two-sided exact binomial test is used to calculate the p-value. The significance level is set at α=0.05, with additional reporting of results at $p<0.01$ and $p<0.001$. Because 15 hypotheses are tested simultaneously on the same test set, all p-values are additionally corrected with the Benjamini-Hochberg procedure [29] described below.

The traditional large-sample McNemar statistic is $\chi^2 = (b - c)^2 / (b + c)$. When the number of discordant pairs is small, this study uses the exact binomial test to compute the two-sided p-value, avoiding bias from the normal approximation:

$$\chi^2 = \frac{(b-c)^2}{b+c},\ p = 2\times\min\{P(B\le b), P(B\ge b)\}$$

Equation (8) McNemar exact binomial test

Where $n = b + c$ is the total number of discordant pairs and $k = \min(b, c)$. If the resulting two-sided p-value is below $\alpha = 0.05$, the null hypothesis is rejected, indicating a statistically significant difference between models A and B on the test set.

When six models are compared pairwise, $m = 15$ tests are performed on the same test set, which inflates the false-positive risk of the whole family. We therefore apply the Benjamini-Hochberg (BH) step-up procedure [29], which controls the false discovery rate (FDR) at $q = 0.05$. Let $p(1) \le p(2) \le \ldots \le p(m)$ denote the ordered raw p-values; the BH-adjusted p-value of the i-th ordered test is:

$$P_{adj}(i) = \min_{j\ge i}\left[\frac{m}{j}p(j)\right]$$

Equation (9) Benjamini-Hochberg adjusted p-value

A pair is considered significant only if its BH-adjusted p-value is below 0.05. As Sections 6.3–6.4 show, this correction changes the conclusions materially: of the five pairs significant at the raw 0.05 level, only BERT-Embedding-TextCNN vs. Qwen3-0.6B and FastText vs. Qwen3-0.6B remain significant after correction.

6. Experimental Results and Analysis

6.1. Accuracy Comparison

Table 4 presents the classification performance of the six models on the 2,118 test samples, with Wilson 95% confidence intervals.

Table 4. Model performance comparison (with Wilson 95% confidence intervals)

Rank	Model	Accuracy	95% CI	Macro F1	Inference Speed	Parameters
1	BERT-Embedding-TextCNN	75.50%	[73.6, 77.3]	75.66%	28,217/s	16.68M total (0.46M trainable)
2	FastText	75.07%	[73.2, 76.9]	75.21%	91,063/s	200.7M alloc. (≈6.8M effective)
3	Qwen3-4B (8-bit)	73.94%	[72.0, 75.8]	74.26%	3.4/s	4B
4	Qwen3-1.7B (FP16)	73.42%	[71.5, 75.3]	73.61%	2.4/s	1.7B
5	BERT-base	73.32%	[71.4, 75.2]	73.31%	121.5/s	102M
6	Qwen3-0.6B (FP16)	71.77%	[69.8, 73.6]	72.07%	24/s	0.6B

In the accuracy ranking, BERT-Embedding-TextCNN and FastText lead at approximately 75% accuracy, Qwen3-4B and Qwen3-1.7B follow (73%–74%), and Qwen3-0.6B ranks last (71.77%). With 2,118 test samples the Wilson 95% confidence intervals span roughly ±1.9 percentage points, so the intervals of adjacent models overlap heavily; rank ordering should therefore be interpreted jointly with the significance tests in Section 6.3. Nevertheless, it is noteworthy that BERT-Embedding-TextCNN, whose trainable component has only 0.46M parameters (16.68M including the frozen embedding table), and FastText, with approximately 6.8M effective parameters (200.7M allocated in its sparse hash table), both match or exceed the accuracy of the 4B-parameter Qwen3-4B.

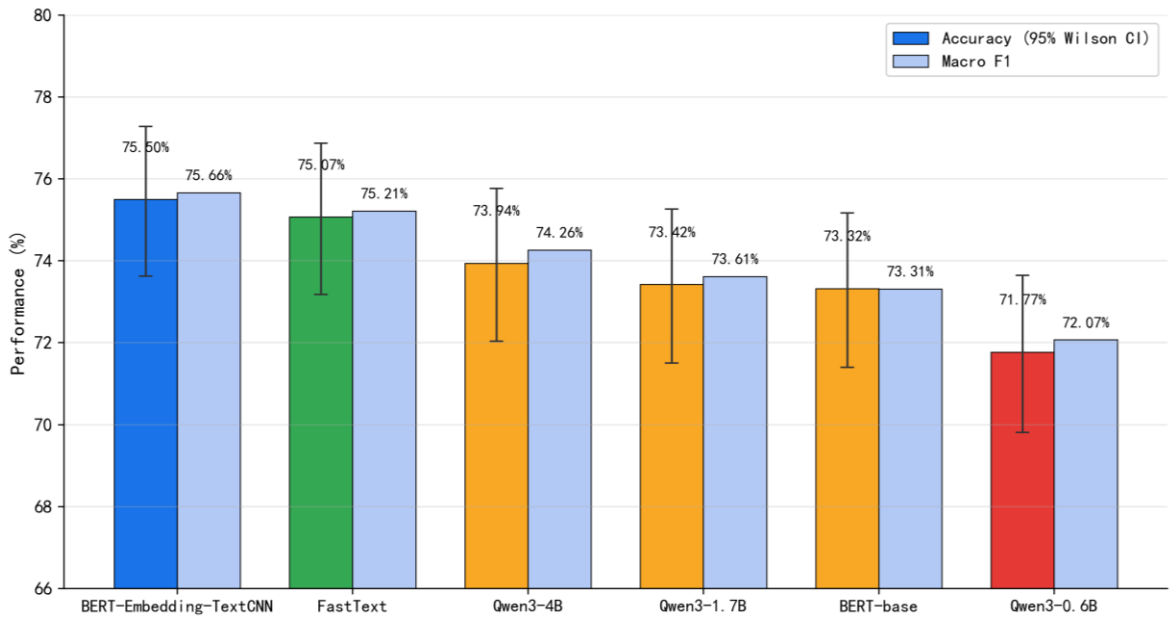

Figure 3. Test accuracy with Wilson 95% CIs and Macro F1 across models

6.2. Per-Class F1 Comparison

To provide an intuitive view of performance differences across the three retention-period categories, Fig. 4 plots the F1 scores of all six models on the permanent, 30-year, and 10-year categories. Compared with dense tables listing precision/recall/F1 row by row, the grouped bar chart more clearly reveals each model's relative strengths and weaknesses across categories.

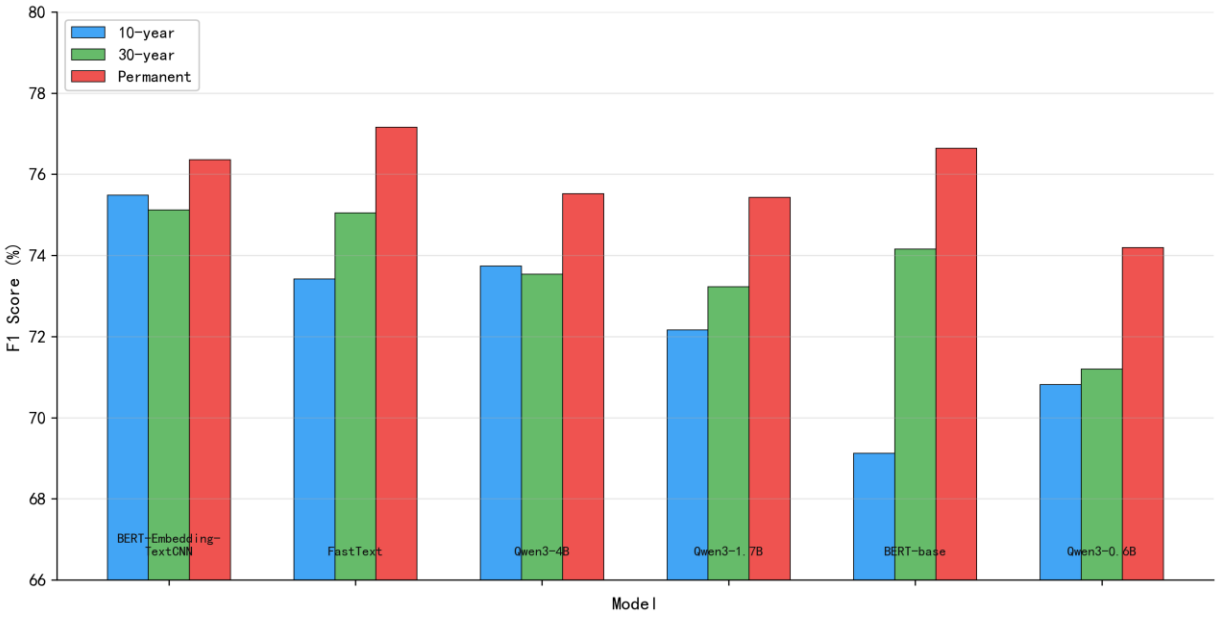

Figure 4. Per-class F1 scores (Qwen3 series computed from exact confusion matrices of mapped generative predictions)

As shown in Fig. 4, FastText achieves the highest F1 for the permanent category (77.16%), BERT-base is relatively weak for the 10-year category (69.12%), and BERT-Embedding-TextCNN is the most balanced across the three categories (76.36/75.12/75.49 on permanent/30-year/10-year). Per-class F1 for the Qwen3 series is computed from the mapped generative predictions on the test set via exact confusion matrices: Qwen3-4B reaches 75.52/73.54/73.74, Qwen3-1.7B 75.43/73.23/72.16, and Qwen3-0.6B 74.19/71.20/70.82 on permanent/30-year/10-year, respectively. All six models find the 10-year category hardest and the permanent category easiest, consistent with the relative class frequencies (25.1% permanent vs. 31.7% 10-year) and with the more heterogeneous wording of routine transactional notices.

6.3. McNemar Significance Test Results

Table 5 presents the p-value matrix of pairwise McNemar tests among the six models; the upper triangle shows raw exact-binomial p-values and the lower triangle shows Benjamini-Hochberg adjusted p-values.

Table 5. McNemar test p-value matrix (upper triangle: raw p; lower triangle: BH-adjusted p; exact binomial test, two-sided)

	BERT-Embedding-TextCNN	FastText	Qwen3-4B	Qwen3-1.7B	BERT-base	Qwen3-0.6B
BERT-Embedding-TextCNN	—	0.6738 ns	0.1117 ns	0.0358*	0.0143*	0.0001***
FastText	0.7219 ns	—	0.2355 ns	0.0926 ns	0.0735 ns	0.0007***
Qwen3-4B	0.1821 ns	0.3212 ns	—	0.5967 ns	0.5669 ns	0.0250*
Qwen3-1.7B	0.1073 ns	0.1821 ns	0.6885 ns	—	0.9599 ns	0.0982 ns
BERT-base	0.0713 ns	0.1821 ns	0.6885 ns	0.9599 ns	—	0.1214 ns
Qwen3-0.6B	0.0016**	0.0056**	0.0939 ns	0.1821 ns	0.1821 ns	—

Note: *** p<0.001, ** p<0.01, * p<0.05, ns=not significant; significance marks apply to the p-value in the corresponding triangle (raw in the upper, adjusted in the lower).

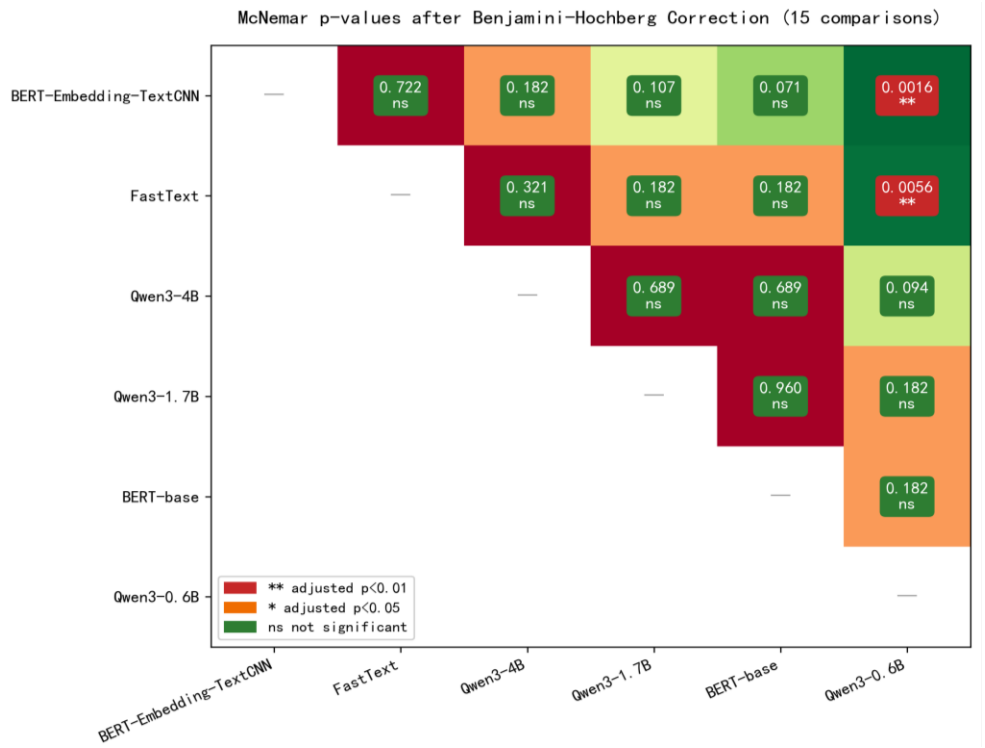

Figure 5. Heatmap of McNemar p-values after Benjamini-Hochberg correction

6.4. Accuracy Ranking and Multiple-Comparison Correction

Based on the raw McNemar results, the six models appear to form three accuracy tiers, as visualized in Fig. 6. After BH correction over the 15 comparisons, however, only two pairwise differences remain statistically significant: BERT-Embedding-TextCNN vs. Qwen3-0.6B (adjusted p=0.0016) and FastText vs. Qwen3-0.6B (adjusted p=0.0056). The raw significance of BERT-Embedding-TextCNN over Qwen3-1.7B (raw p=0.036), of BERT-Embedding-TextCNN over BERT-base (raw p=0.014), and of Qwen3-4B over Qwen3-0.6B (raw p=0.025) does not survive the correction (adjusted p=0.107, 0.071, and 0.094, respectively). The observed three-tier structure is therefore a descriptive clustering of accuracies rather than a set of statistically verified tiers, and the conclusions below are phrased accordingly.

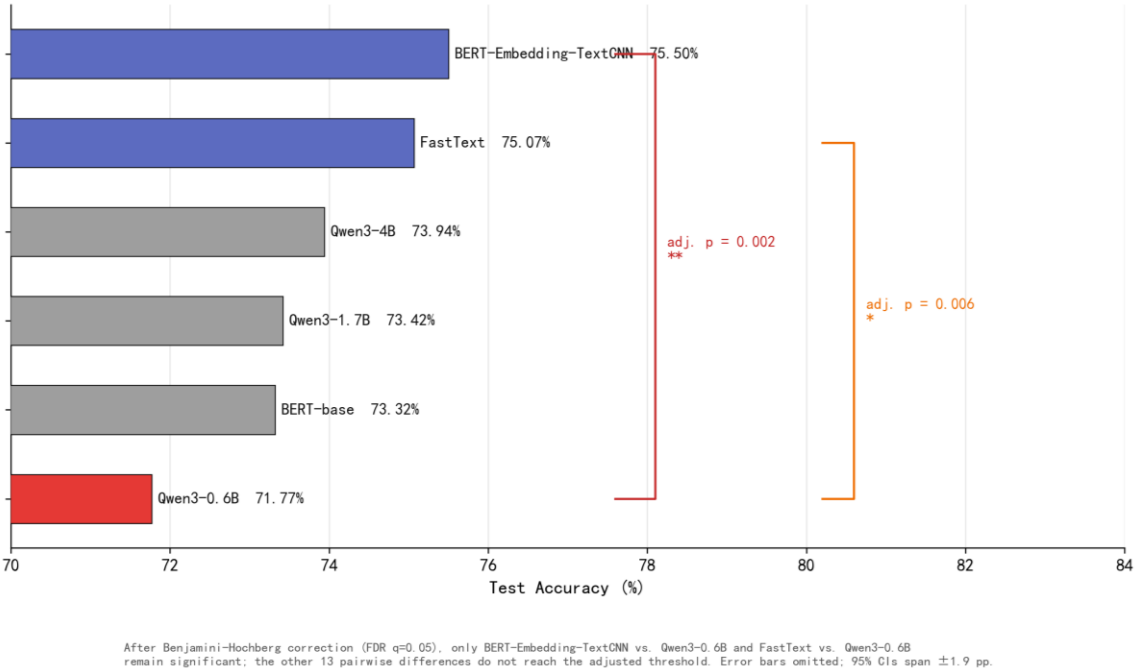

Figure 6. Accuracy ranking and the two differences surviving BH correction

The three highest-accuracy models (approximately 75%) are BERT-Embedding-TextCNN, FastText, and Qwen3-4B. Raw pairwise differences within this group are far above the 0.05 threshold (p=0.24–0.67), and no statistically significant difference was detected among the three models after BH correction. At the raw 0.05 level, BERT-Embedding-TextCNN additionally outperforms Qwen3-1.7B (p=0.036) and BERT-base (p=0.014), but neither difference survives correction (adjusted p=0.107 and 0.071).

The middle group (approximately 73%) comprises Qwen3-1.7B and BERT-base. No statistically significant difference was detected between the two (raw p=0.96, adjusted p=0.96, and an accuracy difference of only 0.10 percentage points); the evaluated 1.7B-parameter LLM and the 102M-parameter BERT-base cannot be distinguished on this test set. Neither significantly outperforms Qwen3-0.6B after correction (adjusted p=0.18 for both).

Qwen3-0.6B (71.77%) is the only model significantly outperformed by multiple models after BH correction: by BERT-Embedding-TextCNN (adjusted p=0.0016) and by FastText (adjusted p=0.0056). Its raw difference against Qwen3-4B (p=0.025) does not survive correction (adjusted p=0.094). Within the evaluated Qwen3 configurations, the 0.6B model shows the lowest accuracy; the evaluated 1.7B and 4B configurations did not differ significantly from each other on this test set after correction.

6.5. Key Findings and Discussion

The keyword-driven classification hypothesis is supported by the results. FastText (75.07%), based on n-gram bag-of-words features, achieves higher accuracy than BERT-base (73.32%), which relies on deep self-attention, although the raw difference (p=0.074) does not reach significance even before correction. This pattern is consistent with the keyword-driven hypothesis: retention-period determination appears to rely heavily on keyword features (e.g., "personnel appointment" → permanent, "general notice" → 10-year) rather than deep semantic reasoning. One possible explanation for BERT-base's underperformance is that its 12-layer self-attention contextualizes and averages character representations, which may dilute sparse keyword signals on short, weakly structured titles; this remains a hypothesis rather than a verified mechanism.

The design of BERT-Embedding-TextCNN is consistent with this explanation: it keeps BERT's pre-trained character embeddings (preserving lexical semantic information) as a frozen lookup, but replaces self-attention with local convolution and global max pooling for feature selection, achieving the best accuracy of 75.50%. The ablation in Table 3 further shows that adding the SE channel-attention module or multi-pooling reduces accuracy, and that randomly initialized embeddings lose 3.3 percentage points; the useful component is the pre-trained character embedding itself, not deeper contextual encoding. Together with the model comparison, this supports the keyword-driven hypothesis.

To move beyond indirect evidence, we inspected the learned FastText feature weights directly. For each high-frequency feature (training frequency ≥ 30), we computed a discriminative margin score—the dot product between the normalized feature embedding and the target label's output vector, minus that of the runner-up label. Table 6 lists the top features per class. They align closely with archival retention rules: the leading permanent-class features—"bidding," "state assets," and "recruitment"—reflect major-decision, asset-disposal, and personnel documents; the leading 30-year-class features—"regulatory plan," "land parcel," and "petition bureau"—reflect planning and petition-response topics; and the leading 10-year-class features—"submission," "training," and "inspection"—reflect routine reporting and training notices. This provides direct, inspectable evidence that archival keywords are important predictive features for the shallow model and supports the hypothesis that lexical cues play a substantial role in task performance.

Table 6. Top discriminative FastText features per retention-period class (training frequency ≥ 30; margin = target-label score − runner-up-label score)

Retention Period	Top discriminative features (gloss, frequency)	Archival interpretation
Permanent	bidding (89); state assets (181); recruitment (34); high-speed rail (52); reform (124); parking (32)	Major decisions, asset disposal, personnel matters
30-year	petition bureau (203); regulatory plan (38); land parcel (211); railway line (1,356); civic inquiry (290); financing (54)	Planning, financing, routine administration
10-year	submission (85); training (37); inspection (70); task assignment (199); EMU operation (82); responsibility letter (34)	Routine transactional notices

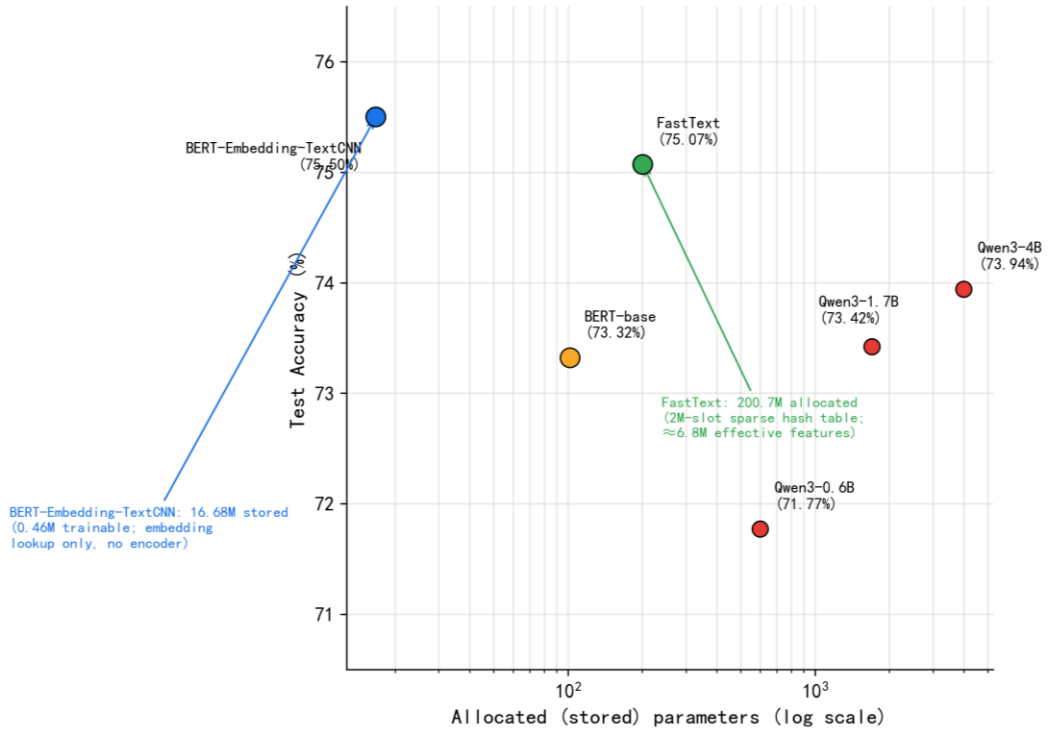

Figure 7. Relationship between allocated (stored) parameters and test accuracy (logarithmic x-axis; parameter accounting differs across architectures—see text)

Across the six evaluated configurations, accuracy did not increase monotonically with model scale. Fig. 7 plots accuracy against the allocated (stored) parameter count on a logarithmic axis; because parameter accounting differs across architectures (allocated, effective, total, and trainable), the x-axis should be read as an indicator of model scale rather than as an isolated experimental variable. FastText (200.7M allocated parameters in its sparse hash table; ≈6.8M effective) and BERT-Embedding-TextCNN (16.68M stored, of which 0.46M trainable) both achieve higher accuracy than Qwen3-4B (4B parameters), although the differences are not statistically significant after correction. Qwen3-0.6B (0.6B parameters) is the lowest-accuracy configuration. This pattern challenges the common assumption that "bigger is better," indicating that on weakly structured text classification tasks, the match between model architecture and task characteristics is more important than parameter scale.

Lightweight models show substantial inference-efficiency advantages under the evaluated deployment configurations. As shown in Table 7, FastText's recorded inference speed is 91,063 samples/s on CPU—roughly 26,800 times that of Qwen3-4B on GPU (3.4/s); the associated accuracy difference is not statistically significant. On the test set of 2,118 samples, Qwen3-4B requires about 620 seconds of GPU generation, while FastText finishes in 0.02 seconds on CPU. This observed throughput ratio reflects both architectural and hardware differences (CPU versus GPU implementations) and should not be interpreted as a hardware-normalized measure of model efficiency. In enterprise CPU-only deployment scenarios, this efficiency gap would be even more pronounced.

Table 7. Inference efficiency comparison of the three highest-accuracy models

Model	Accuracy	Inference Speed	Parameters	Relative Speed
FastText	75.07%	91,063/s	200.7M alloc. (≈6.8M effective)	26,784×
BERT-Embedding-TextCNN	75.50%	28,217/s	16.68M total (0.46M trainable)	8,299×
Qwen3-4B	73.94%	3.4/s	4B	1×

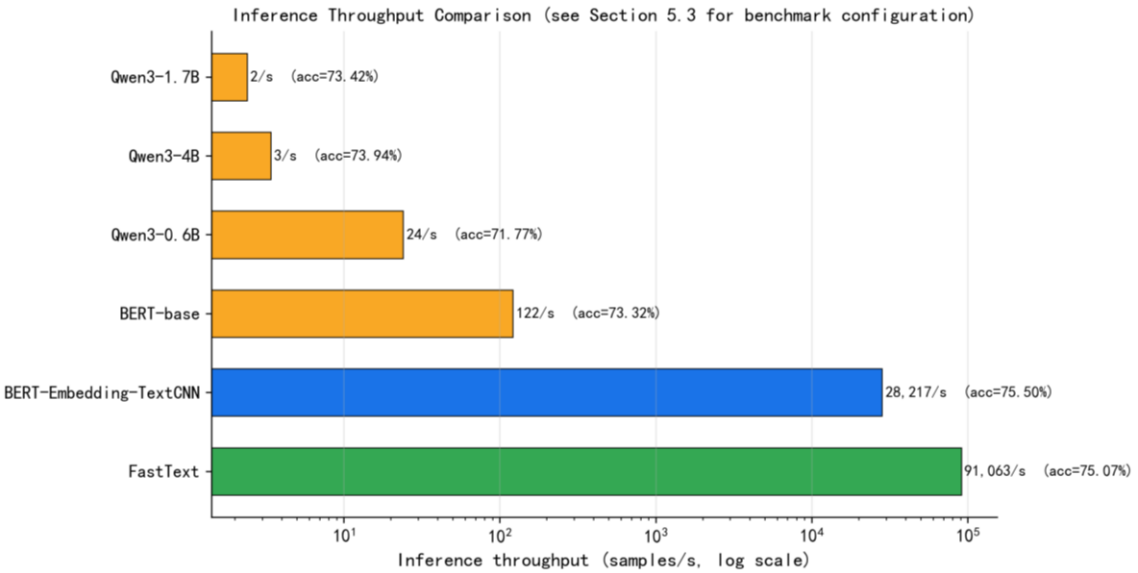

Figure 8. Inference speed comparison (logarithmic scale)

The McNemar test detects no significant difference between BERT-base (73.32%) and Qwen3-1.7B (73.42%) (p=0.96); their accuracies differ by only 0.10 percentage points, while BERT-base's measured inference speed (121.5/s) is approximately 50 times that of Qwen3-1.7B (2.4/s) under the evaluated hardware and inference settings. This finding indicates that on specific text classification tasks, a fine-tuned medium-scale pre-trained language model can match the performance of an evaluated large language model while drastically reducing inference cost.

6.6. Leakage-Controlled Evaluation

To quantify the impact of the train-test overlaps disclosed in Section 5.1, we constructed a leakage-controlled test set by removing the 106 exact duplicates of training documents and the 322 near-duplicates (character-bigram Jaccard similarity above 0.8), retaining 1,690 test documents (79.8% of the original test set). Table 8 reports the accuracy of all six models on this subset, recomputed from their stored per-sample predictions. Three observations follow. First, every model loses 1.7–2.5 percentage points, confirming that the overlaps inflate the reported accuracies, roughly uniformly across architectures. Second, the accuracy ranking is unchanged: BERT-Embedding-TextCNN (73.67%) and FastText (73.08%) remain ahead of Qwen3-4B (71.66%), BERT-base (71.60%), and Qwen3-1.7B (71.01%), with Qwen3-0.6B last (69.23%). Third, re-running the 15 pairwise McNemar tests with BH correction on the leakage-controlled subset leaves both previously significant differences significant—BERT-Embedding-TextCNN vs. Qwen3-0.6B (adjusted p=0.0013) and FastText vs. Qwen3-0.6B (adjusted p=0.0079)—while no other pair reaches significance. The central conclusion—that lightweight keyword-capturing models match or exceed much larger LLMs on this task—is therefore robust to the removal of leaked documents, although the smaller effective test size (n=1,690) widens the confidence intervals to approximately ±2.1 percentage points.

Table 8. Accuracy on the leakage-controlled test set (n=1,690; exact and near-duplicates of training documents removed)

Model	Accuracy (full test, n=2,118)	Accuracy (leakage-controlled, n=1,690)	Δ (pp)	Wilson 95% CI (leakage-controlled)
BERT-Embedding-TextCNN	75.50%	73.67%	-1.83	[71.5, 75.7]
FastText	75.07%	73.08%	-1.99	[70.9, 75.1]
Qwen3-4B	73.94%	71.66%	-2.28	[69.5, 73.8]
Qwen3-1.7B	73.42%	71.01%	-2.41	[68.8, 73.1]
BERT-base	73.32%	71.60%	-1.73	[69.4, 73.7]
Qwen3-0.6B	71.77%	69.23%	-2.54	[67.0, 71.4]

7. Conclusion

Across six models spanning four architectural paradigms, BERT-Embedding-TextCNN (75.50%) and FastText (75.07%) lead the accuracy ranking on weakly structured official-document retention-period classification, ahead of Qwen3-4B (73.94%). After Benjamini-Hochberg correction of the 15 pairwise McNemar tests, however, only two differences are statistically significant: BERT-Embedding-TextCNN and FastText each outperform Qwen3-0.6B (adjusted p=0.002 and p=0.006). The remaining 13 pairwise differences—including all comparisons among the top three models—cannot be distinguished at n=2,118. A leakage-controlled evaluation on the 1,690 test documents that are neither exact nor near-duplicates of training documents preserves this ranking and both significant differences, with all accuracies 1.7–2.5 percentage points lower. The lightweight models' accuracy is thus best described as showing no statistically significant difference from that of the 4B LLM on this test set, while their measured throughput advantage under the evaluated deployment configurations is substantial: FastText delivers 91,063 samples/s on CPU, roughly 26,800× the throughput of Qwen3-4B on GPU, a ratio that reflects hardware as well as architectural differences. Inspection of FastText's feature weights shows that archival keywords are important predictive features for its decisions, supporting a

keyword-driven explanation of task success. For resource-constrained enterprise deployment we recommend FastText first, with BERT-Embedding-TextCNN as an accuracy-oriented alternative; LLM fine-tuning is not justified by the accuracy evidence alone in this setting.

7.1. Limitations and Future Work

This study has several limitations. (1) Single-enterprise dataset: all documents originate from one railway-investment group, and class boundaries follow its internal retention rules; external validity to other industries requires further verification. (2) Near-duplicate leakage: the splits contain exact and near-duplicates of training documents (106 exact train-test duplicates and 322 near-duplicates with Jaccard similarity above 0.8, 20.2% of the test set in total); the leakage-controlled evaluation in Section 6.6 indicates that these overlaps inflate all reported accuracies by roughly 1.7–2.5 percentage points but do not change the ranking or the significant differences. (3) Test-set size: with n=2,118 only differences larger than roughly 2–3 percentage points can reach significance after correction; a larger test set could resolve the currently inconclusive comparisons among the top models. (4) The keyword-driven explanation rests on observational evidence (accuracy patterns and feature weights); controlled keyword-masking experiments are needed to verify the mechanism. (5) LLM inference used greedy decoding with a fixed prompt; prompt optimization or constrained decoding might improve LLM results. (6) The BERT-Embedding-TextCNN architecture is an application of existing ideas (SE-Net, SECNN, BERT-CNN hybrids) rather than a novel architecture. (7) Only Chinese official documents were evaluated; cross-lingual and cross-domain generalization remain untested. (8) Misclassification costs are asymmetric in archival practice—erroneously assigning a permanent document to 10-year retention is less recoverable than the reverse—but the evaluation treats all errors equally. Future work will address keyword-masking experiments, cost-sensitive evaluation, active-learning data collection for boundary samples, and deployment pilots in the enterprise OA system.

Acknowledgements

This study was supported by the Wenzhou Polytechnic University Key Project "Lightweight Electronic Official Document Archive Classification System Based on RoBERTa Chinese Model" (WZY2025011), and received data support from the Archives Department of Wenzhou Railway and Rail Transit Investment Group Co., Ltd.

References

[1] Wang, Q., & Wu, Z. (2020). Integration Framework of Business Systems and Archives Management Systems: Construction and Connotation Analysis. Archives Science Bulletin, (6), 45–53. [In Chinese]

[2] Qu, F. (2024). On Archives Management in Public Institutions. Lantai Inside and Outside, (22), 43–45. [In Chinese]

[3] Han, F. (2024). Research on the Policy Basis and Supply of Data Archiving Based on the 14th Five-Year Plan for National Archives Development. Lantai Inside and Outside, (25), 22–24. [In Chinese]

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017, June 12). Attention Is All You Need. arXiv. https://arxiv.org/abs/1706.03762

[5] General Office of the CPC Central Committee, & General Office of the State Council. (2021, June 9). 14th Five-Year Plan for National Archives Development. https://www.saac.gov.cn/daj/toutiao/202106/ecca2de5bce44a0eb55c890762868683.shtml [In Chinese]

[6] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of Tricks for Efficient Text Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (pp. 427–431).

[7] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (pp. 4171–4186).

[8] Qwen Team. (2025). Qwen3 Technical Report. Alibaba Group.

[9] McNemar, Q. (1947). Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. Psychometrika, 12(2), 153–157.

[10] Public Record Office Victoria. (2020). Victorian Government Email Machine Assisted Appraisal: Proof of Concept. https://prov.vic.gov.au/

[11] The National Archives. (2021, November 8). Using AI for Digital Selection in Government. https://www.nationalarchives.gov.uk/

[12] Pistol, R., & Freise, K. (2024). EHRI Innovation Report (D3.4). King's College London.

[13] European Commission. (2020, June 5). Time Machine Science and Technology Roadmap (D2.2). https://www.timemachine.eu/wp-content/uploads/2020/07/D2.2_TM_CSA_Science_and_Technology_P1_Roadmap_v1.5.pdf

[14] Li, X., & Shu, Z. (2021). Research on Intelligent Archiving Method of Electronic Documents Based on Metadata. Archives, (9), 48–52. [In Chinese]

[15] Kang, Y., & Yuan, J. (2023). Application of "Multi-Agent" Technology in Electronic Document Archiving Management of "One Network Unified Service" in Government Services. China Archives, (4), 64. [In Chinese]

[16] Chen, X. (2022). Artificial Intelligence and Archives Management: Progress, Vision, and Challenges. China Archives, (11), 30–32. [In Chinese]

[17] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (pp. 1746–1751).

[18] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7132–7141).

[19] Yuan, Y., & Zou, Y. (2023). SECNN: A SE-Channel-Based Convolutional Neural Network for Chinese Text Classification. arXiv. https://arxiv.org/abs/2312.06088

[20] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, T., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. In Advances in Neural Information Processing Systems (Vol. 33, pp. 1877–1901).

[21] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023).

LLaMA: Open and Efficient Foundation Language Models. Meta AI.

[22] Zheng, Y. (2024). LLaMA-Factory: Unified Efficient Fine-Tuning of 100+ Language Models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (pp. 1–9).

[23] Ma, Y. B., Cao, Y. X., Hong, Y. C., Li, Y., & Sun, M. (2023). Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples! In Findings of the Association for Computational Linguistics: EMNLP 2023 (pp. 10572–10601).

[24] Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation, 10(7), 1895–1923.

[25] Zhang, Y. F., & Ding, M. (2022). Involved Text Recognition Based on Improved Bidirectional Encoder Representation from Transformers. Science Technology and Engineering, 22(29), 12945–12953. [In Chinese]

[26] Wen, W. Z. H. (2022). News Text Classification System Based on Deep Learning [Master's thesis]. Nanjing University of Posts and Telecommunications. [In Chinese]

[27] Zhang, H., Shan, Y., Peng, J., & Li, C. (2022). A Text Classification Method Based on BERT-Att-TextCNN Model. In Proceedings of the 2022 IEEE International Conference on Mechanical, Electronic, and Information Technology (IMCEC) (pp. 1731–1735).

[28] National Archives Administration of China. (2021). Regulations on the Administration of Enterprise Archives (Order No. 10). https://www.saac.gov.cn/ [In Chinese]

[29] Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society: Series B (Methodological), 57(1), 289–300.