Innovation and Technology Studies
ISSN: 3007-6927
DOI: https://doi.org/10.61784/its3023

LARGE LANGUAGE MODELS FOR ENTERPRISE WORKFLOW AUTOMATION IN FINANCIAL OPERATIONS

BingJie Zi
Northeastern University, Boston MA 02115, USA.

Abstract: Intent classification is a key component of customer-query routing in financial workflow automation. While large language models (LLMs) have attracted substantial interest, their cost, latency, and on-premises deployment constraints motivate efficient routing systems that can operate reliably on standard CPU-based infrastructure. This study examines whether lightweight CPU-based classifiers can serve as a cost-effective routing layer in LLM-era enterprise automation systems. We present a reproducible benchmark on the public Banking77 dataset (13,083 queries, 77 fine-grained intents) comparing five CPU-only pipelines built from term-frequency–inverse-document-frequency (TF-IDF) features and linear classifiers. Our proposed pipeline, LR-Fusion, combines word- and character-level TF-IDF representations within a single logistic regression classifier and attains 0.9123 accuracy and 0.9124 macro-F1 (95% CI [0.9017, 0.9219]) with a median single-query latency of 7.0 ms on a single CPU core. Paired-bootstrap tests indicate that the improvement of LR-Fusion over each baseline is consistent across resamples at the 0.05 significance level. Error analysis identifies semantically overlapping intent pairs (e.g., verify_my_identity vs. why_verify_identity) as the primary residual failure mode, suggesting a potential role for selective LLM-based reranking in future hybrid systems. The implementation package is provided to support reproducibility of the reported experiments.
Keywords: Intent classification; Financial workflow automation; Customer-query routing; TF-IDF; Logistic regression; Banking77; Reproducible baselines; Lightweight NLP

1 INTRODUCTION

Enterprise workflow automation in financial operations spans a wide range of back-office and customer-facing processes, including ticket triage, document categorisation, fraud-alert routing, and conversational customer service. A central recurring task is short-text intent classification: mapping each incoming customer utterance to a canonical workflow category that determines which downstream process or specialist team handles the request. Errors at this routing stage can lead to longer handling times, increased operational cost, and degraded customer experience.

Recent progress on large language models (LLMs) has prompted considerable interest in their use as general-purpose query routers and orchestrators within enterprise automation stacks [1-3]. However, LLM-based routing in financial operations must satisfy three practical constraints that are frequently in tension with model scale: (i) per-query latency budgets in the millisecond range for real-time chat and voice channels; (ii) predictable inference cost under high-volume traffic, where per-call inference fees can accumulate rapidly; and (iii) deployment in on-premises or virtual private cloud environments driven by regulatory and data residency requirements. These constraints make it important to quantify the performance, latency, and deployment trade-offs of lightweight routing systems before introducing LLM-based components.

In this LLM-oriented deployment context, this paper examines whether classical TF-IDF pipelines can serve as an efficient first-stage routing layer for financial workflow automation. We provide a fully reproducible, CPU-only benchmark on the public Banking77 dataset that quantifies the accuracy [4], latency, training cost, and per-class behaviour of five lightweight pipelines, ranging from a naive Bayes baseline to a fused word- and character-level logistic regression classifier. Bootstrap confidence intervals and paired-bootstrap significance tests accompany every reported metric. We also provide a fine-grained error analysis that isolates the small set of semantically overlapping intent pairs that account for a substantial portion of the remaining errors and that could be selectively re-routed to an LLM-based reranker in a production pipeline. All code, data-download scripts, and result tables are released to facilitate reproducibility of the reported results.

The remainder of the paper is organised as follows. Section II reviews relevant work on intent detection and financial NLP. Section III describes the task formulation and the five pipelines. Section IV details the experimental protocol. Section V presents results, statistical comparisons, and error analysis. Section VI concludes and outlines directions for hybrid LLM-augmented extensions.

2 RELATED WORK

Intent classification has been studied extensively as a foundational task for task-oriented dialogue and customer-service automation. The Banking77 dataset established a fine-grained single-domain benchmark with 77 banking intents and has since been widely adopted for evaluating both classical and neural classifiers [4]. Complementary multi-domain benchmarks include CLINC150 [5], which introduces out-of-scope queries spanning 150 intents over ten domains. Both benchmarks have been used to compare classical TF-IDF baselines against sentence-encoder-based models.

Pre-trained language models built on the Transformer architecture have become a common starting point for many text classification systems [6]. Encoder-only models such as BERT introduced bidirectional masked language modelling and produced strong fine-tuning baselines across a range of NLP tasks [7]. Decoder-only language models such as GPT-3 subsequently demonstrated that very large models can perform competitive few-shot classification through prompting [1]. Retrieval-augmented generation (RAG) connects parametric LLMs with non-parametric document indices and has become a common pattern in enterprise deployment of LLMs [2]. Although these approaches are powerful, they introduce model-size, latency, and infrastructure overheads that are not always justified for high-volume short-text routing tasks.

Domain-adapted models for finance include FinBERT variants [8,9], which further pre-train BERT on financial corpora and report improvements on sentiment-analysis tasks. BloombergGPT is a 50-billion-parameter LLM trained on a mixed financial and general corpus [3]. While these models offer general-purpose capability across a spectrum of financial NLP tasks, lightweight task-specific classifiers remain attractive for high-throughput routing tasks.

Within classical text categorisation, linear classifiers operating over sparse bag-of-n-gram representations remain competitive on short-text tasks. Joachims established that support vector machines are well suited to the high-dimensional [10], sparse feature spaces produced by TF-IDF, and subsequent practice has favoured logistic regression for its calibrated probabilistic outputs and ease of training with libraries such as scikit-learn [11]. We build on this line of work and present the resulting pipelines as an efficient CPU-based routing solution and a reference point for future hybrid LLM-assisted systems.

3 METHOD

3.1 Task Formulation

Given a short customer-service utterance x represented as a sequence of tokens over an alphabet Σ, the task is to predict a single intent label y from the set {1, ..., K}, where K = 77 for Banking77. The objective is to maximise macro-averaged F1 on a held-out test set while keeping per-query inference latency low enough to support real-time routing.

3.2 Feature Representations

We use two complementary TF-IDF representations. The first is a word-level vectoriser with unigrams and bigrams, a minimum document frequency of two, sublinear term-frequency scaling, and lower-cased text. The second is a character n-gram vectoriser operating on the character_wb mode of scikit-learn, capturing n-grams of length three to five that respect word boundaries. Character n-grams are known to compensate for morphological variation, typographical noise, and partial out-of-vocabulary tokens that are common in user-entered customer-service queries.

3.3 Pipelines

We evaluate five CPU-only pipelines that share the same train/test interface:

1) NB-Word: word TF-IDF → multinomial naive Bayes. This is a standard baseline for sparse high-dimensional text data.

2) LR-Word: word TF-IDF → multinomial logistic regression with the L-BFGS solver. We use C = 4.0 and a maximum of 2000 iterations.

3) SVM-Word: word TF-IDF → linear support vector machine with C = 1.0 [10].

4) LR-Char: character n-gram TF-IDF → logistic regression with the same hyperparameters as LR-Word.

5) LR-Fusion (proposed): horizontal concatenation of word- and character-level TF-IDF matrices, fed to a single logistic regression classifier. This design combines the morphological robustness of character features with the lexical specificity of word-level features, and avoids the additional complexity of training and combining separate ensemble components.

All pipelines are implemented with scikit-learn and are deterministic under a fixed random seed [11].

3.4 Evaluation Protocol

We report accuracy, macro-averaged F1, weighted F1, training time, batch prediction time on the full test set, and median single-query latency measured under a warm-up of fifty queries followed by five hundred timed single-query predictions. We compute 95% bootstrap confidence intervals for macro-F1 using one thousand resamples of the test set. For the proposed pipeline we also report paired-bootstrap two-sided p-values against each baseline; the test statistic is the difference in macro-F1, and the null distribution is centred on zero.

4 EXPERIMENTS

Experiments are run on the public Banking77 dataset [4], using the canonical train/test split distributed by the dataset authors. The training set contains 10003 examples and the test set contains 3080 examples (forty per class). All 77 intent labels are represented in both splits. Text is used as provided, with only leading and trailing whitespace stripped.

All experiments are run on a single x86-64 CPU core at 2.8 GHz with approximately 4 GB of available memory inside

a containerised Linux environment, using Python 3.12, scikit-learn 1.8, NumPy 2.4, and SciPy 1.17. No GPU is required at any point in the pipeline. Absolute latency numbers should be interpreted as indicative rather than as portable across hardware. The reproducibility package includes a data-download script that pulls the dataset directly from the official GitHub repository without authentication, the experiment driver, and the figure-generation code. All randomness is controlled by a single seed (42).

5 RESULTS AND DISCUSSION

5.1 Main Results

Table 1 reports test-set accuracy, macro-F1 with 95% bootstrap confidence intervals, weighted F1, and median single-query latency for all five pipelines. The naive Bayes baseline reaches 0.7990 accuracy and 0.7857 macro-F1, while the two word-level linear classifiers (LR-Word and SVM-Word) improve macro-F1 to 0.8885 and 0.8906 respectively. The character n-gram classifier LR-Char further improves macro-F1 to 0.9059, indicating that sub-word morphological cues are useful for short banking queries. The proposed LR-Fusion pipeline achieves the best macro-F1 of 0.9124 (95% CI [0.9017, 0.9219]).

Table 1 Test-set Performance on Banking77 (3,080 examples, 77 classes). 95% Bootstrap CIs Computed from 1,000 Resamples

Model	Accuracy	Macro-F1	95% CI	Weighted-F1	Latency (ms)
NB-Word	0.7990	0.7857	[0.771, 0.798]	0.7857	1.50
LR-Word	0.8880	0.8885	[0.876, 0.898]	0.8885	1.46
SVM-Word	0.8906	0.8906	[0.879, 0.901]	0.8906	0.57
LR-Char	0.9058	0.9059	[0.895, 0.916]	0.9059	2.16
LR-Fusion	0.9123	0.9124	[0.902, 0.922]	0.9124	7.01

Figure 1 visualises the same comparison across accuracy, macro-F1, and weighted F1. Macro-F1 and weighted F1 are nearly identical for each model because the official test split is balanced at forty examples per class.

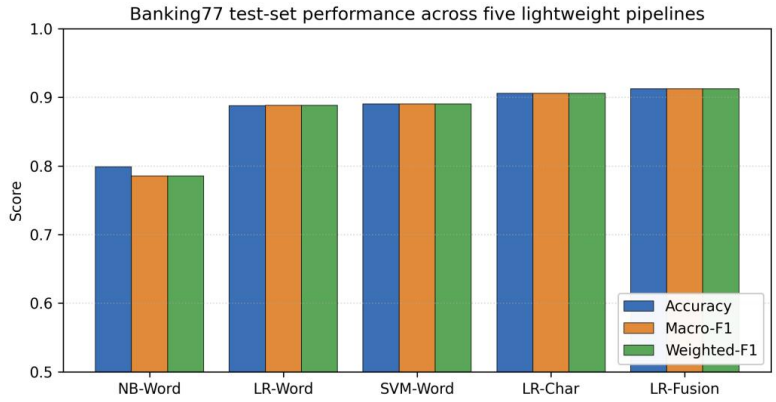

Figure 1 Test-set Performance of the Five Lightweight Pipelines on Banking77

5.2 Statistical Significance

Table 2 reports paired-bootstrap two-sided p-values comparing LR-Fusion to each of the four alternatives. The improvement of LR-Fusion is significant at the 0.05 level against every baseline. The gain over LR-Char is numerically incremental at 0.0066 macro-F1, but it is consistent under paired-bootstrap resampling ($p = 0.017$) and is obtained through a simple feature-level fusion strategy rather than additional model capacity.

Table 2 Paired-bootstrap Two-sided p-values (1,000 resamples) for the Proposed LR-Fusion Pipeline Versus each Alternative

Comparison	Δ Macro-F1	p-value
LR-Fusion vs NB-Word	0.1267	0.001
LR-Fusion vs LR-Word	0.0240	0.001
LR-Fusion vs SVM-Word	0.0219	0.001
LR-Fusion vs LR-Char	0.0066	0.017

5.3 Latency and Training Cost

Table 3 summarises training time, full-test batch prediction time, and median single-query latency. SVM-Word offers the lowest per-query latency at 0.57 ms with very modest training cost; LR-Fusion is the most expensive of the lightweight pipelines at 7.01 ms per query on the test hardware, which remains suitable for many real-time routing settings where millisecond-level latency is required. The full Pareto front of accuracy versus latency is visualised in Fig. 2; LR-Fusion occupies the high-accuracy, higher-latency end of the trade-off curve.

Table 3 *Training, Full-test Batch Prediction, and Single-query Latency on one CPU Core*

Model	Fit time (s)	Batch predict (s)	ms / query (median)
NB-Word	0.25	0.054	1.50
LR-Word	9.19	0.055	1.46
SVM-Word	0.88	0.051	0.57
LR-Char	13.60	0.182	2.16
LR-Fusion	20.23	0.246	7.01

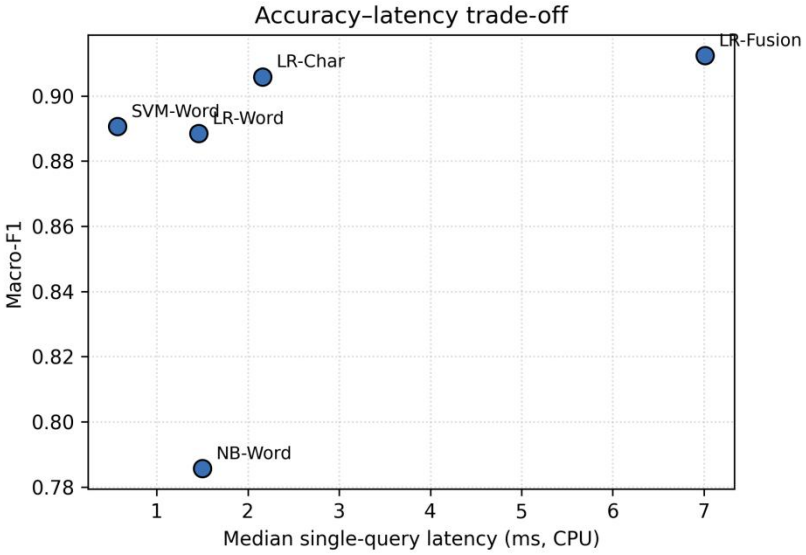

Figure 2 Accuracy–latency Trade-off across the Five Pipelines

5.4 Per-Class Behaviour and Error Analysis

Fig. 3 shows the distribution of per-class F1 for LR-Fusion across all 77 intents. The median per-class F1 is 0.914, with a minimum of 0.753 and a maximum of 1.000. Of the 77 intents, 49 (i.e., 63.6%) reach per-class F1 ≥ 0.90 and only 9 fall below 0.85. The classifier therefore delivers near-balanced quality across the workflow categories rather than relying on easy classes to inflate the macro average.

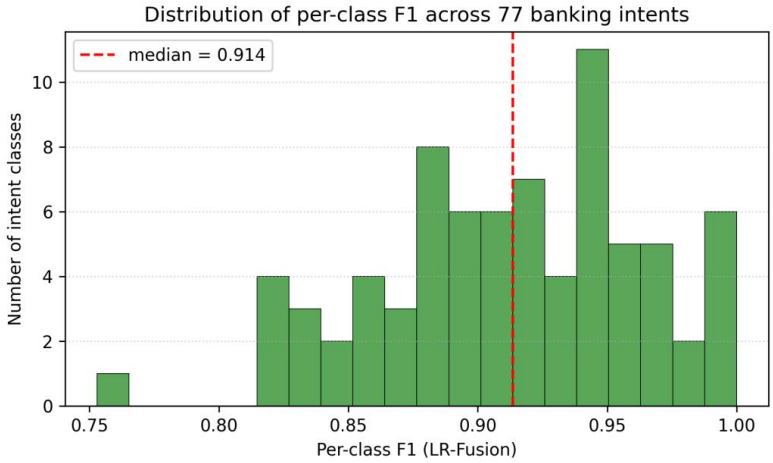

Figure 3 Distribution of Per-class F1 for LR-Fusion across 77 Banking Intents

Examining the off-diagonal entries of the confusion matrix in Table 4 reveals that the dominant errors involve semantically overlapping intent pairs. The pair (verify_my_identity, why_verify_identity) accounts for five test-set errors in each direction, and (top_up_failed, top_up_reverted) accounts for similarly bidirectional confusion. These

pairs are inherently difficult because their distinction often depends on contextual information (e.g., whether the user is asking why a process exists versus asking whether a specific instance succeeded). Figure 4 visualises the top confused pairs, which collectively account for a large share of the residual errors. This pattern suggests a practical hybrid design in which the lightweight classifier handles the high-confidence majority of traffic and routes only this narrow band of semantically ambiguous queries to an LLM-based reranker, limiting LLM-related inference cost while preserving the latency advantages of the lightweight stage.

Table 4 Top Confused Intent Pairs under LR-Fusion (Seven most Frequent Misclassifications)

True intent	Predicted intent	Count
virtual_card_not_working	get_disposable_virtual_card	5
verify_my_identity	why_verify_identity	5
top_up_failed	top_up_reverted	4
why_verify_identity	unable_to_verify_identity	4
balance_not_updated_after_bank_transfer	transfer_not_received_by_recipient	4
card_acceptance	card_not_working	3
pin_blocked	card_not_working	3

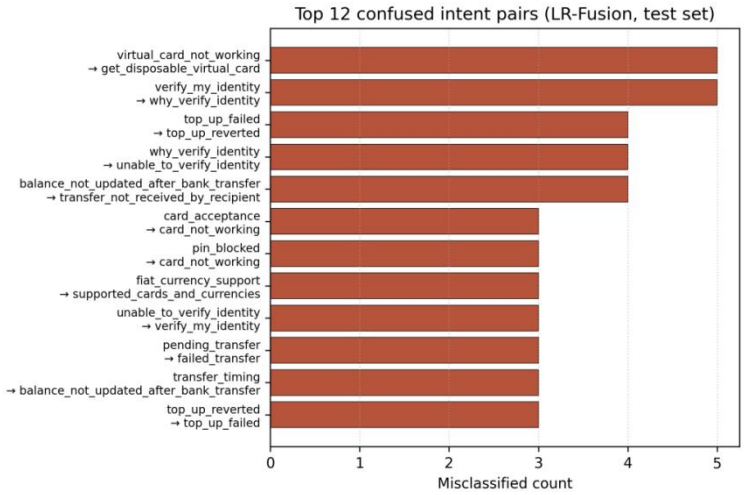

Figure 4 Most Frequent Confused Intent Pairs under LR-Fusion on the Banking77 Test Set

5.5 Discussion

The results are consistent with the longstanding observation that on short, single-utterance text classification tasks, well-tuned sparse linear classifiers remain a strong reference point. The gap between word-only and word+character fusion indicates that future extensions, such as domain-adapted contextual encoders [8], or retrieval-based augmentation [2,9], should be evaluated not only by accuracy but also by their latency, cost, and deployment implications. For production financial-operations workflows, the pipeline reported here provides a useful default: it is reproducible, runs entirely on CPU, has predictable latency, and produces probabilistic outputs that downstream confidence-based routing logic can consume directly.

6 CONCLUSION

We presented a focused, reproducible benchmark of five CPU-only TF-IDF pipelines for intent classification on the public Banking77 dataset, framed as a deployment-oriented routing solution for LLM-era enterprise workflow automation in financial operations. The proposed LR-Fusion pipeline, which combines word and character n-gram features within a single logistic regression classifier, achieves the strongest accuracy and macro-F1 among the lightweight pipelines considered, with median single-query latency that remains suitable for many real-time routing settings. Paired-bootstrap comparisons show that the gains over each alternative are consistent across resamples, and the per-class error analysis identifies a small set of semantically overlapping intent pairs as the dominant residual failure mode. Future work may extend this system by evaluating selective reranking for low-confidence or semantically overlapping queries, adding out-of-scope detection following CLINC150-style protocols, and testing transferability to other financial workflows such as complaint triage and document categorisation.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Brown T B, Mann B, Ryder N, et al. Language models are few-shot learners. Advances in Neural Information Processing Systems (NeurIPS), 2020, 33: 1877-1901.
[2] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems (NeurIPS), 2020, 33: 9459-9474.
[3] Wu S, Irsoy O, Lu S, et al. BloombergGPT: A large language model for finance. arXiv preprint, 2023. DOI: 10.48550/arXiv.2303.17564.
[4] Casanueva I, Temčinas T, Gerz D, et al. Efficient intent detection with dual sentence encoders. Proc. 2nd Workshop on Natural Language Processing for Conversational AI (ACL), 2020: 38-45.
[5] Larson S, Mahendran A, Peper JJ, et al. An evaluation dataset for intent classification and out-of-scope prediction. Proc. EMNLP-IJCNLP, Hong Kong, China, 2019: 1311-1316.
[6] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS), 2017, 30: 5998-6008.
[7] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. Proc. NAACL-HLT, Minneapolis, MN, USA, 2019: 4171-4186.
[8] Araci D. FinBERT: Financial sentiment analysis with pre-trained language models. arXiv preprint, 2019. DOI: 10.48550/arXiv.1908.10063.
[9] Yang Y, Uy M C S, Huang A. FinBERT: A pretrained language model for financial communications. arXiv preprint, 220. DOI: 10.48550/arXiv.2006.08097.
[10] Joachims T. Text categorization with support vector machines: Learning with many relevant features. Proc. 10th European Conference on Machine Learning (ECML), LNCS, Springer, 1998, 1398: 137-142.
[11] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 2011, 12(85): 2825-2830.