On-Device Classification of Canadian Bank Transaction Descriptions

A Research and Implementation Report

Author: Maaz Zaidi
Student Number: 300246507
Period of Work: March 17 to April 6, 2026
Benchmark: 505 unique transaction descriptions extracted from 114 RBC bank statement PDFs (2019 - 2026), representing 3,113 weighted occurrences

---

GitHub Repository:
https://github.com/Maaz-Zaidi/Transaction-Classifier

Preface

I'll apologize in advance, this report is meant to be long by design. The project it documents is not a single experiment with a clean outcome; it is twelve phases of iteration, each used as a basis for the ones before it, that allowed me to learn by result. Compressing that process would strip the document of the very thing that makes it useful: being a record of how this problem was researched, analyzed, tackled, and improved upon. The length reflects the work, not the prose.

This project was also motivated by something broader than a course deliverable. To the best of my knowledge, no open-source, privacy-preserving financial transaction categorizer exists that is built for real Canadian bank data. Most people have no easy way to understand where their money actually goes. The long-term goal of this work is to change that: to build something that runs entirely on a person's own device, requires no subscription, and makes financial accountability genuinely accessible to anyone, anywhere in the world. That is the ceiling this project is working toward.

I want to acknowledge two forms of assistance that shaped this work. First, I want to sincerely thank the professor for granting an extension on this submission. The additional time was not a courtesy; it was the difference between a project that stopped at a shallow proof-of-concept and one that reached a competitive result and understood why. The extra days produced Phases 5 through 12, which represent the majority of the technical insight in this document.

Second, I want to credit OpenAI. Their tools assisted with portions of the implementation code across several phases, and ChatGPT was used to help label the significantly expanded transaction dataset introduced in Phase 5 (described in detail in Section 3.2). All ChatGPT-assisted labels were reviewed personally, with additional spot-checking from family members, before being accepted into the benchmark. OpenAI's assistance meaningfully accelerated this work and that contribution deserves acknowledgment, as otherwise, it would have been extremely tedious.

Abstract

This report documents the design, implementation, and evaluation of a fully local machine learning pipeline for classifying raw Canadian bank transaction descriptions into ten personal finance categories. The work spanned twelve phases over eleven days, progressing from a baseline TF-IDF classifier through sentence transformers, an external merchant knowledge base, dense retrieval with cross-encoder reranking, token-aware query decomposition, and enriched model retraining. The final system reached 73.9% unique accuracy (83.6% weighted) on a held-out benchmark of 505 real Canadian bank transactions, up from an initial 43.4%. Each phase is documented with its hypothesis, implementation, result, and the diagnostic reasoning that motivated the next. The central finding is that the primary bottleneck in transaction classification is not model architecture or training scale, but the availability of diverse, domain-matched merchant names in training data, and the quality of external merchant entity resolution.

Table of Contents

1. Introduction and Problem Statement

Personal finance applications require the ability to map raw bank transaction descriptions to meaningful budget categories. A string such as `CONTACTLESS INTERAC PURCHASE - 1234 TIM HORTONS KANATA ON` must resolve to "Food & Dining," while `Payroll Deposit ERICSSON CANADA` must resolve to "Income." The task is complicated by the fact that real bank statements are noisy, abbreviated, institutionally inconsistent, and contain a mixture of merchant names, bank-specific prefixes, location suffixes, reference numbers, and terminal identifiers. No standardized format exists across Canadian financial institutions, and the same merchant can appear under dozens of different descriptor patterns on different statement types.

The goal of this project was to build a classification service that (a) runs entirely on-device with no transaction data leaving the machine, (b) operates on consumer hardware (a phone, or CPU based computer), (c) classifies transactions into a ten-category taxonomy, and (d) serves predictions through a FastAPI endpoint suitable for integration with a personal budgeting application.

The ten target categories were adopted from the mitulshah/transaction-categorization dataset on HuggingFace [1]: Food & Dining, Transportation, Shopping & Retail, Entertainment & Recreation, Healthcare & Medical, Utilities & Services, Financial Services, Income, Government & Legal, and Charity & Donations. An earlier fifteen-category schema was abandoned because sufficient training data for the additional categories (Groceries, Subscriptions, E-Transfers, Banking, Education, Miscellaneous) did not exist in any public dataset.

The core technical challenge, identified early and confirmed repeatedly, is the domain gap between available synthetic training data and real Canadian bank transaction descriptions. No public dataset of labeled Canadian bank transactions was identified at any point during this work. The best available training resource, the mitulshah dataset with 3.6 million records, is synthetically generated with clean, US-centric merchant names that bear little resemblance to the truncated, noisy, and Canada-specific strings found on actual bank statements.

2. Related Work

2.1 Industry Production Systems

Every production transaction classification system identified in the literature review uses a hybrid architecture combining deterministic rules with machine learning, layered with an external merchant knowledge base.

Plaid [2] processes 500 million transactions daily. Their system combines fuzzy matching for known merchants, a BERT-style masked language model pretrained on their transaction corpus, a BiLSTM for merchant name extraction, and SafeGraph Places data covering six million points of interest with Merchant Category Codes. In 2025, Plaid introduced a contrastive-learning transaction foundation model that improved income classification by 48%, loan payment detection by 14%, and bank fee classification by 22%.

Ntropy [3] has processed 100 million unique merchants and reports that the top 500 merchants cover 50% of all transactions, but the remaining 50% is a long tail of millions of smaller merchants. Rules-based approaches or zero-shot large language models alone typically achieve only 60-70% accuracy without a supporting merchant database.

Brex [4] built a two-stage system where the Google Places API identifies the merchant from the descriptor text and location, and an ML classifier operates on the resolved entity. Google Places resolves approximately 50% of merchants; the combined system achieves an overall error rate well below 5%.

The universal pattern across these systems is that entity resolution comes first. Once the merchant identity is known, the category is largely deterministic. The merchant knowledge base, not the ML model, is the primary classification mechanism. My research arrived at the same conclusion through iterative improvements.

2.2 Academic Literature

A 2025 paper on SME transaction classification [5] used GPT-4o to generate synthetic transaction descriptions and fine-tuned FinBERT with focal loss, achieving 73.4% standard accuracy and 90.4% high-confidence accuracy. Zero-shot GPT-4o alone reached only 60.4%, underscoring that even state-of-the-art language models require domain adaptation for transaction classification.

A study on French Open Banking transactions [6] trained on 94,356 records across 84 categories; Word2Vec with Random Forest achieved 95% F1, and TF-IDF with Linear SVM achieved 94%. Research on weakly supervised bank transaction classification [7] found that domain-specific FastText embeddings outperformed character CNNs, off-the-shelf embeddings, and other classical approaches.

A Federal Reserve paper on active knowledge distillation [8] proposed using Gemma-3-27B as a teacher model, achieving up to 80% reduction in labeling requirements. A comparison of small models versus LLMs [9] found that a 1.7-million-parameter model achieved near-identical accuracy to Llama3-8B (72.07% vs. 72.89%) at one-tenth the cost and eight times the speed. The BTZSC benchmark [10] established that zero-shot classification remains significantly below fine-tuned models, with the best zero-shot result at 0.72 macro F1.

2.3 Open-Source Projects

Five open-source projects were reviewed for architectural patterns. Common findings: every successful project starts with rules and adds ML later; transaction amount is an underused disambiguation feature; user correction loops are the primary long-term differentiator; 10-15 categories is the practical ceiling; and preprocessing accounts for roughly half of total development effort. These patterns held throughout this project.

2.4 Datasets

The mitulshah/transaction-categorization dataset [1] was selected as the primary training resource: 4.5 million records across five countries including Canada, with ten categories. The Foursquare OS Places dataset [11] (Apache 2.0, 100M+ global points of interest) was identified as the primary external merchant knowledge resource; the Canada subset contains 2.38 million entries with hierarchical category labels up to six levels deep. No public dataset of labeled Canadian bank transactions was found.

3. Dataset and Evaluation Protocol

3.1 Training Data

All models were trained on the mitulshah dataset. The full dataset contains 4,501,043 records, reduced to 4,497,324 after removing empty descriptions, split 80/10/10 stratified by category. Training subsets ranged from 8,000 to 173,761 samples depending on the phase. The dataset is nearly perfectly balanced at approximately 10% per category.

A deep analysis in Phase 12 revealed a structural limitation that recontextualized all earlier results: the 3.6 million rows contain only **847 unique base merchant or concept names**. The remaining rows are generated by crossing these names with country markers, time-of-day markers, location suffixes, transaction codes, and payment methods. Training on more rows beyond roughly 10,000-20,000 samples yields no additional information. Row count is not a proxy for information content; effective diversity is measured by unique base entities.

3.2 Evaluation Data

The evaluation benchmark was expanded progressively across the project. The initial benchmark for Phases 1 through 4 comprised 370 transactions extracted from 20 RBC PDFs. All category labels for this set were assigned manually by me, reviewing each transaction description against the ten-category taxonomy. While this was time-intensive, it allowed me to produced a clean initial ground truth that served as the foundation for all early-phase evaluation.

In Phase 5, the benchmark was significantly expanded to 114 RBC PDFs covering both MasterCard and chequing accounts from 2019 to 2026, producing 3,113 total transaction descriptions and 505 unique descriptions after deduplication on cleaned text. Manually labeling this volume in full was not practical within the project timeline. ChatGPT was used to generate an initial set of category assignments across the full 505 descriptions. All suggested labels were then reviewed personally against the taxonomy, and a further round of spot-checking was conducted with the help of younger siblings to catch obvious errors or ambiguous assignments. **To note,** I recognize that LLM-assisted labeling introduces its own biases and cannot be treated as ground truth with the same confidence as fully manual annotation. The 505-row benchmark is best understood as a high-quality approximation. To bound the noise, I also cross-validated against Gemini's labels on overlapping descriptions showed 93.3% agreement, suggesting the

benchmark is reliable for tracking relative improvements across phases, even if individual labels carry some uncertainty. The rest were reviewed personally.

3.3 Preprocessing Pipeline

A text preprocessing pipeline was developed and refined iteratively across the project:

- **Direction detection** on raw, unpreprocessed text: payroll deposits, e-transfer autodeposits, and government payments are identified and routed before any text modification, achieving near-100% accuracy through deterministic matching.
- **Bank prefix stripping:** removes standardized patterns including "Contactless Interac purchase –", "Visa Debit purchase –", "Online Banking payment –", "Misc Payment –", "ATM withdrawal", "Mobile cheque deposit", and Toast POS "TST-" prefixes.
- **Refund prefix handling:** strips "REFUND –" prefixes while preserving the merchant name, preventing refund transactions from being misclassified as Income.
- **E-transfer direction tagging:** differentiates incoming and outgoing e-transfers using keyword matching on raw text.
- **Location suffix stripping:** removes trailing Canadian city and province patterns.
- **Noise removal:** strips reference numbers, card numbers, long digit sequences, and special characters, preserving &, apostrophes, periods, hyphens, and slashes that appear in legitimate merchant names.

Example: `CONTACTLESS INTERAC PURCHASE - 1234 TIM HORTONS KANATA ON` becomes `TIM HORTONS`.

4. Phase 1: Baseline, TF-IDF and SGD Classifier

Date: March 23, 2026

Hypothesis: Character-level TF-IDF features with an SGD classifier would provide a fast, CPU-trainable baseline with strong performance on structured transaction text.

4.1 Implementation

The baseline model used a TfidfVectorizer with character word-boundary n-grams (range 3-5), 100,000 maximum features, and sublinear TF weighting, feeding into an SGDClassifier with modified Huber loss and balanced class weights. Training time was 60.9 seconds on CPU over 3.6 million samples. A rules engine was implemented with approximately 60 regex and keyword patterns covering known Canadian merchants.

4.2 Results

On the synthetic validation set, the SGD model achieved 98% accuracy with 0.94 average confidence, and the ensemble (rules + SGD) reached 95.7% on the synthetic test set at 30,468 transactions per second. On real bank data (370 manually labeled transactions from 20 RBC PDFs), accuracy collapsed to **43.4%**. Rules accuracy dropped from 80.5% to 77.2%, while SGD accuracy fell from 98.3% to **22.3%**, a 76-point regression from the synthetic benchmark. The 52.3-point gap between synthetic and real accuracy established the central challenge of the project.

4.3 Failure Analysis

- **PDF extraction failure:** pdfplumber stripped spaces in RBC PDFs, producing joined strings such as "TIMHORTONS #1664KANATAON". Approximately 35-40% of descriptions were affected, breaking both preprocessing and n-gram matching.
- **Domain mismatch:** the SGD model had never seen real Canadian merchant names (RCSS, SHAWARMA PALACE, QUICKIE, RIDEAU BOURBON ST). 80.3% of real-world vocabulary was absent from training.
- **Preprocessing gaps:** several bank prefixes found in real statements (Contactless Interac, Online Banking payment, TST-) were not handled by the initial pipeline.

5. Phase 1b: PDF Extraction Fix

Date: March 25, 2026

Replacing pdfplumber with pymupdf (fitz) improved overall accuracy from 43.4% to **53.7%** (+10.3 points). MasterCard accuracy improved from 58.2% to 67.3%. Chequing account accuracy remained essentially flat (32.8% -> 32.5%), confirming that chequing failures were driven by domain mismatch and preprocessing gaps, not spacing errors. The fix recovered approximately half the accuracy lost to extraction artifacts, while leaving the fundamental vocabulary gap unaddressed.

6. Phase 2: Direction Detection and FastText

Date: March 26, 2026

Hypothesis: A two-stage architecture; a) deterministic direction detection followed by b) FastText subword character n-grams; would address both the income classification gap and the truncated merchant name problem.

6.1 Implementation

Stage 1 applied deterministic prefix matching on raw text to classify payroll deposits, direct deposits, e-transfer autodeposits, and government tax credits immediately. Stage 2 combined the rules engine with a FastText supervised classifier trained on 3.6 million samples over 10 epochs using subword character n-grams of length 3-6, dimension 100, word bigrams, and softmax loss. Training time: 125 seconds on CPU.

6.2 Results

Overall accuracy improved from 53.7% to 55.7%. Income rose from 2.2% to **97.8%**, and Government & Legal rose from 0.0% to 54.5%. Chequing account accuracy jumped from 32.5% to 58.2%. However, FastText exhibited a strong Income bias for unknown merchants, predicting Income at 0.90+ confidence for restaurants and stores that it had not seen in training. MasterCard accuracy dropped from 67.3% to 54.1%.

6.3 Analysis

Direction detection was the single highest-impact architectural change to that point, achieving 100% accuracy on its predictions. FastText, despite its theoretically stronger subword handling, shared the same domain mismatch as SGD: trained on synthetic data, its character n-grams for Income-category patterns overlapped with proper noun merchant names, producing systematic misclassification of restaurants and retailers as Income.

7. Phase 3: SetFit with Sentence Transformers

Date: March 28, 2026

Hypothesis: A pretrained sentence transformer fine-tuned with SetFit's contrastive learning framework would leverage pretraining knowledge of real-world concepts to overcome the domain mismatch.

7.1 Implementation

SetFit [12] was applied to all-MiniLM-L6-v2 (22M parameters) on 8,000 stratified samples with contrastive pair generation (1 epoch, batch size 32, 20 iterations producing 320,000 contrastive pairs). Training time: 54 minutes on CPU.

7.2 Results

Overall accuracy improved from 55.7% to **80.5%**, a gain of 24.8 points, the largest single-phase improvement in the project. ML-only accuracy rose from 14.8% to 66.7%. Food & Dining improved by 38.5 points (45.4% -> 83.9%). Financial Services improved by 56.1 points (35.1% -> 91.2%).

7.3 Analysis

The result confirmed the hypothesis: pretrained sentence transformer embeddings encode semantic knowledge of real-world concepts from their pretraining on one billion sentence pairs. MiniLM already knows that "shawarma" and "sushi" relate to food, and that "payment" and "credit" relate to finance. This semantic grounding eliminated the Income bias that afflicted FastText and SGD. Domain transfer through pretraining partially substituted for in-domain labeled examples, which is the key insight of this phase.

8. Phase 4: Standard Fine-Tuning of MiniLM

Date: March 30, 2026

Hypothesis: With 800 samples per class, standard cross-entropy fine-tuning should outperform SetFit, which was designed for few-shot scenarios with 8-16 examples per class.

The same all-MiniLM-L6-v2 base was fine-tuned using AutoModelForSequenceClassification with the HuggingFace Trainer (3 epochs, batch size 64, learning rate 2e-5, warmup ratio 0.1, max token length 64). Training time: 10 minutes on CPU, 5.4x faster than SetFit. An experiment with 20,000 samples achieved 99% synthetic validation accuracy but lower real-world accuracy (80.3%), confirming that additional synthetic data beyond a certain threshold increases overfitting rather than generalization. 8,000 samples proved to be the optimal trade-off.

Overall accuracy improved from 80.5% to **84.5%**. ML-only accuracy improved from 66.7% to 75.1% (+8.4 points). Utilities & Services accuracy doubled from 34.2% to 68.4%. This phase concluded the initial model development arc at an apparent 41.1-point improvement from baseline (a result that looked strong until evaluation was expanded in Phase 5).

9. Phase 5: Full Corpus Evaluation

Date: April 2, 2026

The test set was expanded from 20 to 114 RBC PDFs, covering MasterCard and chequing accounts from 2019 to 2026. Total extracted descriptions: 3,113. After deduplication: 505 unique descriptions. Labels were generated with ChatGPT assistance and reviewed manually, as described in Section 3.2.

On the expanded benchmark, accuracy dropped to **65.3% unique / 72.7% weighted**. ML-only accuracy fell to 58.5%. The 33.9-point gap between synthetic validation (92.4%) and real-world ML performance (58.5%) recalibrated expectations. The expanded corpus surfaced patterns that the smaller test set had masked: abbreviated merchant codes (WMT SUPRCTR, CDN TIRE, MACS CONV) were destroyed by WordPiece tokenization; Canadian-specific merchants (CHATIME, DOLLARAMA, UNIV OF OTTAWA) were entirely absent from training; bilingual descriptions (PAIEMENT, REMBOURSEMENT) had no representation. Model confidence was nearly uncorrelated with prediction accuracy. This 505-row set became the definitive benchmark for all subsequent phases.

10. Phase 6: Abbreviation Augmentation and CANINE

Date: April 2-3, 2026

Hypothesis: The ML accuracy gap was caused by WordPiece tokenization destroying abbreviated merchant names. Two interventions were proposed: augmenting training data with synthetic abbreviations, and replacing the tokenizer with a character-level model (CANINE).

10.1 Augmented MiniLM

50,000 base samples were augmented with three abbreviation variants each (vowel removal, consonant compression, truncation, acronyms), producing 173,761 training examples. Overall accuracy regressed from 65.3% to **52.5%**, and ML-only accuracy fell from 58.5% to 41.2%. The model was confidently wrong (0.90 average confidence at 41% accuracy).

10.2 CANINE Character-Level Model

Google's CANINE-s [13] (132M parameters, character-level transformer) was trained on the same augmented data for 5 epochs. Accuracy regressed further to **39.4%** overall, 23.7% ML-only, at 0.97 average confidence, maximum confidence, minimum accuracy.

10.3 Pivot

Both interventions failed. The hypothesis was wrong: the bottleneck was not tokenization. MiniLM's sentence-level pretraining provided better semantic understanding than CANINE's character-level pretraining, even with imperfect tokenization of abbreviations. The augmented abbreviation patterns did not match how real Canadian banks truncate descriptions, and the preprocessing pipeline already handled most real-world truncation upstream. This phase produced the critical reframe: the problem was not "how to build a better classifier" but "how to resolve merchant identity." Every production fintech system in the literature reaches this same conclusion.

11. Phase 7: External Merchant Knowledge Base

Date: April 4, 2026

Hypothesis: Adding external merchant knowledge would improve accuracy by resolving merchant identity before classification.

11.1 Implementation

A merchant knowledge base (KB) was constructed from the Foursquare OS Places Canada subset (2.38 million rows scanned, 132 entries matched to test-set merchants) and 32 manually curated entries for high-frequency Canadian merchants and digital services. The merged KB contained 164 entries. At runtime, the pipeline operated in three modes: direct KB resolution for high-confidence matches, metadata enrichment for plausible matches, and raw ML fallback for unmatched transactions.

A note on methodology: the initial KB was constructed using the local transaction corpus as the candidate name set for Foursquare lookup. **To note,** before this is misinterpreted, this is not label leakage (the builder never read category labels), but Phase 7 results do not represent a strict holdout of unseen merchants. I was experimenting with a deliberate scaffold to validate the KB concept before committing to a full corpus build. Phase 8 corrected this by ingesting the complete Canada Foursquare corpus independently of the evaluation set.

11.2 Results

Configuration	Unique Acc.	Weighted Acc.	KB Direct Hits	Notes
No KB (baseline)	65.4%	72.7%	N/A	ML only
Curated-only KB (32 entries)	68.3%	81.1% (biased)	N/A	Manual curation
Foursquare-only KB	70.1%	75.6%	N/A	Automated
Merged KB (Phase 7)	71.9%	83.1%	116 @ 96.6%	Best Phase 7 result
Dense retrieval (Phase 8)	73.9%	83.6%	100 @ 97.0%	1.8M-entry store

Direct KB hits achieved 96.55% accuracy on 116 predictions, the most precise component of the entire pipeline across all phases. The improvement over no-KB baseline was statistically

significant (McNemar $p \approx 5.4 \times 10^{-7}$). The bottleneck shifted from corpus coverage to retrieval quality.

12. Phase 8: Dense Retrieval and Cross-Encoder Reranking

Date: April 4-5, 2026

Hypothesis: Replacing the small candidate-filtered KB with full dense retrieval and cross-encoder reranking over the complete Canada Foursquare corpus would improve merchant recall.

12.1 Architecture

- **SQLite knowledge store:** 1,795,842 merchant entries with aliases and FTS5 full-text search. Artifact size: 2.4 GB.
- **Chroma persistent collection:** dense vector index using BAAI/bge-small-en-v1.5 [14] (33M parameters, 384 dimensions). Artifact size: 7.1 GB.
- **Hybrid retrieval:** dense (embedding) and lexical (BM25/FTS5) retrieval in parallel, with candidate fusion in application code.
- **Cross-encoder reranking:** BAAI/bge-reranker-v2-m3 [14] (568M parameters, multilingual) scoring the top 20-50 fused candidates.
- **Decision gate:** high-confidence matches routed to direct KB category; plausible matches to metadata-enriched ML; weak matches to raw ML fallback.

12.2 Results

Overall accuracy: **73.86% unique, 83.62% weighted** (up from 71.88%/83.14%). KB direct: 100 predictions at 97.0%. Fine-tuning with metadata: 200 predictions at 67.5%. Fine-tuning only: 76 predictions at 40.8%. The full corpus eliminated coverage as the primary bottleneck; retrieval quality was now the limiting factor.

Remaining failures traced to retrieval quality issues: lexical retrieval was too permissive with flat OR queries; dense retrieval was skewed by location-heavy metadata ("ZARA BAYSHORE" drifted toward Bayshore-area venues rather than the Zara clothing brand); candidate fusion over-weighted semantically related but factually incorrect entries.

13. Phase 9: Rules Overhaul and Token-Aware Query Decomposition

Date: April 5, 2026

13.1 Rules Overhaul

All approximately 60 merchant-identity rules (TIM HORTONS, COSTCO, AMAZON, WALMART, UBER, NETFLIX, etc.) were removed. These rules had served as scaffolding that allowed overall accuracy to remain competitive while the KB and ML components matured. I intended them as a temporary measure, even though I recognize that retaining merchant-identity rules alongside a merchant KB creates a maintenance liability. Rules cannot distinguish COSTCO GAS from COSTCO GROCERY from COSTCO WHOLESALE, the KB can. Thus, Rules were limited to their proper scope: approximately 25 structural patterns that unambiguously indicate a category regardless of the merchant (MORTGAGE, LOAN PAYMENT, CREDIT CARD PAYMENT, NSF, PAYROLL, CRA, CANADA REVENUE, SERVICE ONTARIO, and generic descriptors such as PHARMACY and DENTAL).

13.2 Token-Aware Query Decomposition

A token analyzer module was created to decompose cleaned transaction strings into typed tokens: brand tokens (COSTCO, TIM, HORTONS), descriptor tokens mapping to category-semantic concepts (GAS -> "gas station, fuel"; PHARMACY -> "pharmacy drugstore"), location tokens (OTTAWA, KANATA, BAYSHORE), and noise tokens (branch identifiers, terminal IDs, digit sequences). This decomposition fed cleaner queries into all three retrieval paths and injected descriptor context into ML inputs even when no KB match was found.

13.3 Results

Overall accuracy: 71.3% (down from 73.86%). The 2.56-point drop was the expected cost of retiring the scaffolding rules. KB direct: 161 predictions at 91.3% (up from 100 at 97.0%). The pipeline now operated in its intended final architecture. The gap between Phase 8 and Phase 9 represented the work remaining for the KB and ML to fully absorb the responsibilities of the retired rules.

14. Phase 10: Category Mapping Overhaul and Match Quality Gating

Date: April 5, 2026

14.1 Analysis Findings

The KB matched 504 of 505 test transactions (99.8% retrieval coverage). Of those 504 matches: 274 had a correct mapped category, 52 had a wrong mapped category, and **178 had no mapped category at all**. A structural bug in the Foursquare category mapper caused the `any()` function to short-circuit after the first keyword match, producing tied confidence scores that were then rejected by the confidence threshold. "Retail > Grocery Store" produced a 0.50/0.50 tie between Food and Shopping and was discarded.

14.2 Three Fixes

Category mapper redesign: per-keyword scoring with specificity weighting. Generic keywords (RETAIL, STORE, SHOP) received weight 1.0; specific category-identifying keywords (GROCERY, PHARMACY, GAS STATION, RESTAURANT) received weight 2.0. The confidence threshold was lowered from 0.55 to 0.40. 384,030 entries in the SQLite store were updated, with 326,544 gaining a category for the first time.

Match quality gating: dense-lexical matches had only 28% category accuracy compared to 65% for reranked matches. A strategy-aware gate required dense-lexical matches to meet similarity ≥ 0.80 for metadata enrichment; below that threshold, the ML received raw text rather than potentially wrong metadata.

Descriptor override: for multi-purpose merchants (COSTCO, AMAZON, SHOPPERS DRUG MART), when the token analyzer's descriptor context mapped to a specific category conflicting with the KB category, the descriptor category took precedence. Example: "AMAZON WEB SERVICES" with descriptor "cloud computing, technology services" overrode the KB's Shopping category to produce Utilities & Services.

14.3 Results

Overall accuracy: **72.5%** (+1.2 points). KB direct: 172 predictions at 94.8%. Fine-tune with metadata: 156 at 68.6%. Fine-tune only: 155 at 48.4%. The quality gate shifted 84 transactions from the metadata path to the raw ML path, preventing wrong metadata from corrupting ML inputs.

15. Phase 11: Enriched Model Retraining

Date: April 6, 2026

Hypothesis: The fine-tuned MiniLM was trained on raw merchant names but receives enriched text at inference (merchant name + Foursquare category path). Retraining on enriched-format inputs would allow the model to use the KB metadata signal.

15.1 Methodology

An initial attempt to enrich training data by looking up synthetic merchant names in the KB failed: 86.6% of synthetic merchants matched wrong KB entries, and only 38.2% of all matches had correct categories. Training on wrong metadata would teach the model to trust unreliable signals.

The solution was to sample real Foursquare label paths from the KB for each training sample's known category. For a Food & Dining training sample, a genuine label such as "Dining and Drinking > Restaurant > Fast Food Restaurant" was sampled from the pool of labels that correctly map to Food & Dining. This preserved format fidelity without introducing incorrect metadata. The clean label pool contained 851 unique labels across 9 categories.

Continuation training from the existing checkpoint was critical: a model trained from scratch on enriched data achieved only 70.1%, it learned to expect metadata and failed when metadata was absent. Optimal hyperparameters: 20,000 samples, 50% enrichment ratio, learning rate 5e-6 (one quarter of the original), 2 epochs, with 8% of enriched samples receiving deliberately wrong metadata to build robustness against retrieval errors.

15.2 Results

Best configuration: **73.9% accuracy** (373/505), up from 72.5%. Fine-tune with metadata path: 153 predictions at **77.1%** (was 68.6%, +8.5 points). Fine-tune only path: 155 predictions at 44.5% (was 48.4%, -3.9 points). The metadata path improvement was the core gain; the raw-text path regressed slightly due to mild catastrophic forgetting from continuation training.

16. Phase 12: Deep Diagnostic Analysis

Date: April 6, 2026

16.1 Root Cause: Training Data Poverty

The 3.6-million-row mitulshah dataset contains only 847 unique base merchant names. Per non-Income category, there are approximately 41-62 base names, each expanded to ~560 variants. Of the 505 test merchants, only **15 (3.0%)** had an exact base-name match in training data, and 3 of those 15 were assigned to the wrong category. 97% of test merchants were entirely absent from training. Vocabulary overlap between training and test was 12.0%. These figures explain why augmenting to 173,761 samples performed no better than the 8,000-sample baseline: the dataset was information-exhausted.

16.2 Root Cause: Foursquare Taxonomy Mismatch

The Foursquare taxonomy categorizes by business type (Retail > Pharmacy), not by transaction purpose. The enrichment training taught the model that "Retail > X" means Shopping, conflicting with the project's mapping where "Retail > Pharmacy" indicates Healthcare. The model learned an incorrect taxonomy from the raw Foursquare labels.

16.3 Root Cause: KB Retrieval Errors

Several metadata errors traced to wrong merchant variant matches: LOBLAWS matched to Loblaws Garden Centre; CDN TIRE STORE matched to Q Stop CDN Tire Gas; UBER CANADA/UBEREATS matched to Uber Canada HQ. The rerank strategy was particularly prone to returning loosely related entries with wrong metadata when confidence was borderline.

17. Summary of Results Across Phases

Phase	Date	Key Change	Unique Acc.	ML-Only Acc.	Notes
1	Mar 23	TF-IDF + SGD baseline	43.4%	22.3%	pdfplumber extraction
1b	Mar 25	pymupdf PDF fix	53.7%	28.4%	+10.3 pts from extraction fix
2	Mar 26	Direction detection + FastText	55.7%	14.8%	Income 2% -> 98%; FastText income bias
3	Mar 28	SetFit (MiniLM contrastive)	80.5%	66.7%	Pretrained embeddings: +24.8 pts
4	Mar 30	MiniLM standard fine-tune	84.5%	75.1%	Best result on 497-sample eval
5	Apr 2	Full corpus evaluation	65.3%	58.5%	Expanded eval reveals true gap
6	Apr 3	Abbrev. augmentation + CANINE	52.5% / 39.4%	41.2% / 23.7%	Both regressed; hypothesis disproven
7	Apr 4	External merchant KB	71.9%	N/A	KB direct hits at 96.6% accuracy
8	Apr 5	Dense retrieval + cross-encoder	73.9%	N/A	Full 1.8M-entry store indexed
9	Apr 5	Rules overhaul + token analyzer	71.3%	N/A	Scaffolding removed; intended arch.
10	Apr 5	Category mapper + quality gate	72.5%	N/A	Descriptor override added
11	Apr 6	Enriched model retraining	73.9%	N/A	Metadata path +8.5 pts
12	Apr 6	Deep diagnostic analysis	N/A	N/A	Root causes identified; future roadmap

The chart below visualizes the accuracy trajectory. Green bars indicate improvement over the previous phase; red bars indicate regression. The orange bar (Phase 5) marks the benchmark expansion, which reset the baseline on a harder, more representative evaluation set. The dashed trend line traces the overall direction of progress.

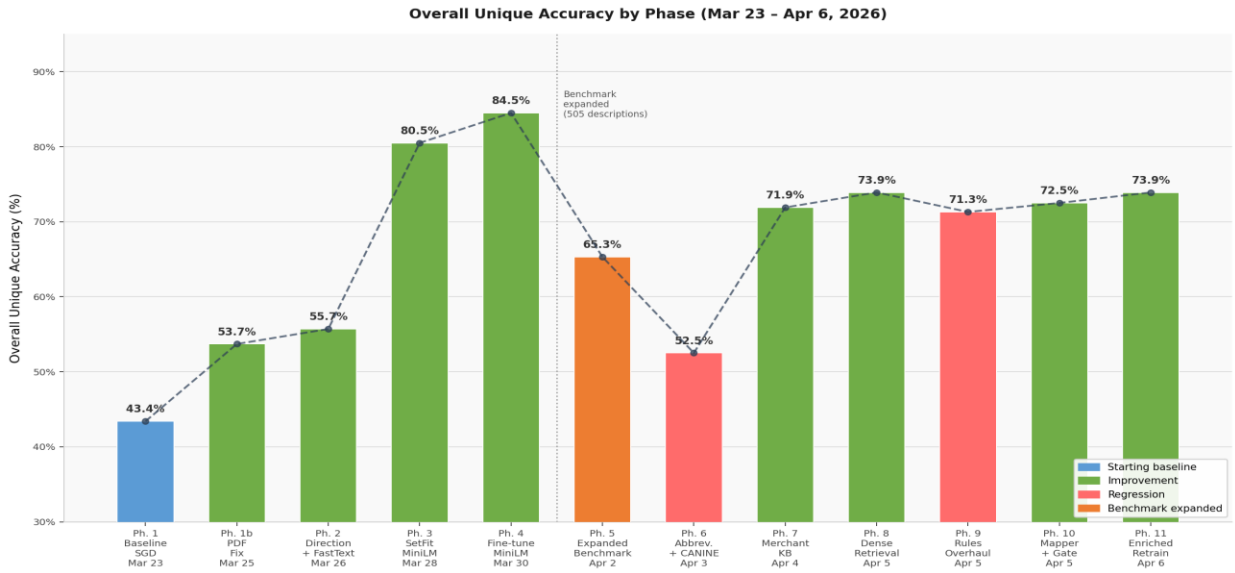

Figure 1. Overall unique accuracy by phase, March 23 – April 6, 2026.

Accuracy on the 505-row benchmark progressed from 65.3% (Phase 5 baseline) to 73.9% (Phase 11), an 8.6-point gain through architectural improvements without any new training data. On a weighted basis (accounting for transaction frequency), the final system achieved 83.6%. The weighted metric better reflects real-world utility: high-frequency merchants such as Tim Hortons, Shoppers Drug Mart, and Starbucks are classified correctly even when their first occurrence drives the denominator for the unique metric.

18. Discussion

18.1 Entity Resolution as the Primary Problem

The most significant finding of this project is that transaction classification is primarily an entity resolution problem, not a text classification problem. Once the merchant identity is established (RCSS -> Real Canadian Superstore -> Food & Dining), the category assignment is deterministic. This aligns with every production system reviewed: Plaid [2], Ntropy [3], and Brex [4] all resolve merchant identity first and classify second. The project arrived at this conclusion empirically through the failure of Phase 6 and the success of Phase 7. The KB direct resolution path achieved 94-97% accuracy across Phases 7-11, consistently the most precise component of the pipeline.

18.2 The Synthetic Data Trap

The mitulshah dataset, despite its 3.6 million rows, is informationally equivalent to approximately 847 labeled examples. The template expansion creates an illusion of scale that masks fundamental vocabulary poverty. This trap was invisible from synthetic validation metrics (98-99% accuracy) and only became apparent on real data (58.5% ML-only accuracy). Practitioners working with synthetic transaction data should measure effective diversity by unique base entities, not total record count.

18.3 Pretrained Embeddings as Domain Transfer

The single largest accuracy gain (Phase 3, +24.8 points) came from switching to a pretrained sentence transformer. No architectural innovation, data augmentation strategy, or external knowledge source produced a comparable improvement in a single step. Knowledge of real-world concepts encoded during pretraining on one billion sentence pairs carried directly into the classification task without any domain-specific examples.

18.4 The Preprocessing Investment

Data preprocessing was a continuous thread across nearly every phase. The initial PDF extraction failure (Phase 1b), missing bank prefix patterns (Phase 2), refund prefix handling (Phase 9), and location suffix stripping all required iterative refinement. The preprocessing pipeline handled Canadian-specific patterns (Interac, e-transfer, bilingual descriptions) that no general-purpose NLP tool addresses. Preprocessing was not a one-time setup cost but an ongoing investment through the final phase.

18.5 Limitations

- **Single institution:** all evaluation data comes from RBC. Generalization to TD, CIBC, BMO, Scotiabank, and Desjardins is untested.
- **Ground truth quality:** ChatGPT-assisted labeling, reviewed manually and with family spot-checking, introduces its own biases and cannot be treated as gold-standard annotation. Some category assignments are genuinely ambiguous (Shoppers Drug Mart: healthcare or shopping? COSTCO: food or retail?). The benchmark includes these cases without resolution.
- **Transductive KB construction:** the Phase 7 KB was built using merchant names from the evaluation corpus as the candidate list. Label leakage was avoided, but results do not represent a strict unseen-merchant holdout. Phases 8 onward corrected this.
- **No deployment evaluation:** all accuracy figures are offline benchmark results. Latency under load, throughput, and user satisfaction in a real budgeting application are unmeasured.
- **No active learning:** a planned self-correction UI and incremental retraining from user corrections remain unimplemented.

19. Future Work and Continuation

This project is not finished. It began with a simple observation: there is no open-source, privacy-preserving financial transaction categorizer built for real Canadian bank data. Most personal finance tools that offer categorization are either cloud-dependent, subscription-based, US-centric, or all three. The long-term goal of this work is a system that anyone in the world can run on their own hardware, for free, in any language their bank uses. That goal has a high ceiling, and the current system is well-positioned to approach it.

The Phase 12 analysis identified a clear and achievable improvement roadmap. Conservative estimates place achievable accuracy at 78-85% with near-term interventions, and 85-91% with the full roadmap. The theoretical upper bound, accounting for genuine label ambiguity, is estimated at 92-95%.

Step	Solution	Expected Gain	Effort	Cumulative Est.	Rationale
12A	Fix Foursquare label mapping + expand rules	+4%	Low	~78%	Corrects taxonomy mismatch in enrichment training
12B	KB entries as distant supervision	+4%	Medium	~82%	847 training names -> 50,000+ real merchants
12C	LLM-generated Canadian training data	+3%	Medium	~85%	Realistic truncated / bilingual / noisy names
12D	Contrastive learning for KB retrieval	+4%	Medium	~89%	Domain-specific entity resolution embeddings
12E	Two-stage entity resolution pipeline	+2%	Low (builds on 12D)	~91%	Mirrors Brex / Plaid production architecture

The most impactful near-term action is replacing the synthetic training data. The 847-name training vocabulary is the single largest bottleneck in my opinion. LLM-generated Canadian merchant names combined with the 1.3 million KB entries as distant supervision would expand the effective training vocabulary from 847 to over 50,000 real merchant names, a two-order-of-magnitude increase in data diversity at modest cost.

The longer-term step, contrastive learning for entity resolution, following the approach used by Plaid's transaction foundation model and the Eridu project [15], would address the fundamental retrieval quality problem. A domain-specific embedding model trained on KB alias pairs would understand that "CDN TIRE" and "CANADIAN TIRE" refer to the same entity, that "UBEREATS" and "UBER EATS" are the same service, and that "MACS CONV" is a Mac's Convenience Store. This is the key gap between the current system and production-grade accuracy.

The addition of multi-institution support, active learning from user corrections, and transaction amount features each represent independent avenues for improvement. Together, the roadmap traces a realistic path to a system that is not only technically competitive with commercial alternatives, but that is free, local, and available to everyone.

20. References

Datasets

[1] mitulshah/transaction-categorization. HuggingFace. https://huggingface.co/datasets/mitulshah/transaction-categorization

[11] Foursquare OS Places. HuggingFace. https://huggingface.co/datasets/foursquare/fsq-os-places

Banking77. HuggingFace. https://huggingface.co/datasets/PolyAI/banking77

greggles/mcc-codes. GitHub. https://github.com/greggles/mcc-codes

Academic Papers

[5] "Categorising SME Bank Transactions with Machine Learning and Synthetic Data Generation." arXiv:2508.05425.

[6] "Specialized Text Classification for Open Banking Transactions." arXiv:2504.12319.

[7] "Weakly Supervised Bank Transaction Classification." arXiv:2305.18430.

"Hierarchical Financial Transaction Classification." arXiv:2312.07730.

[10] "BTZSC: Zero-Shot Text Classification Benchmark." arXiv:2603.11991.

[8] "Active Knowledge Distillation for Transaction Classification." arXiv:2511.11574.

[9] "Better with Less: Small Models vs. LLMs for Financial Transactions." arXiv:2509.25803.

[13] Clark et al. "CANINE: Pre-training an Efficient Tokenization-Free Encoder for Language Representation." TACL 2022.

"BPE-Dropout: Simple and Effective Subword Regularization." Provilkov et al., ACL 2020.

"Tokenization Falling Short." ACL Findings 2024.

"Comparing BERT Against Traditional ML for Text Classification." arXiv:2005.13012.

Industry and Engineering References

[2] Plaid. "Building a Transaction Foundation Model for Intelligent Finance." https://plaid.com/blog/building-transaction-foundation-model-intelligent-finance/

Plaid. "AI-Enhanced Transaction Categorization." https://plaid.com/blog/ai-enhanced-transaction-categorization/

Plaid. "Transaction Enrichment Engine." https://plaid.com/blog/transaction-enrichment-engine/

[4] Brex. "How We Built a Mostly Automated System to Solve Credit Card Merchant Classification." https://medium.com/brexeng/

[3] Ntropy. "Transaction Categorization." https://www.ntropy.com/blog/transaction-categorization

Ntropy. "Understanding Financial Transactions." https://www.ntropy.com/blog/understanding-financial-transactions

Meniga. "Transaction Categorisation Guide." https://www.meniga.com/resources/transaction-categorisation/

Triqai. "Why Transaction Categorization Fails." https://www.triqai.com/article/why-transaction-categorization-is-hard

Capital One. "Imputing Merchant Information Using Sequence Classification." https://capitalone.com/tech/machine-learning/

FreeAgent. "Fine-Tuning BERT for Multiclass Categorisation." https://engineering.freeagent.com/2021/09/15/

Models and Tools

[12] SetFit. HuggingFace. sentence-transformers/all-MiniLM-L6-v2.

[14] BAAI/bge-small-en-v1.5; BAAI/bge-reranker-v2-m3. HuggingFace.

MoritzLaurer/deberta-v3-base-zeroshot-v2.0. HuggingFace.

google/canine-s. HuggingFace.

Open-Source Projects

[15] Graphlet-AI/eridu. HuggingFace. Contrastive learning for company name entity matching.

j-convey/BankTextCategorizer. GitHub.

robintw/BankClassify. GitHub.

eli-goodfriend/banking-class. GitHub.

Foxel05/Finance-TransactionCategorizer. GitHub.

GlenCrawford/bank_transaction_unsupervised_clustering. GitHub.

Standards

Visa Merchant Data Standards Manual. Visa Inc.

Plaid Personal Finance Category Taxonomy. https://plaid.com/documents/transactions-personal-finance-category-taxonomy.csv

Foursquare Places Data Schema. https://docs.foursquare.com/data-products/docs/places-os-data-schema