Republic Of Tunisia
Ministry Of Higher Education And Scientific Research
University Of Tunis
Tunis Business School

Explainable Open-Source License Classification:

Comparative Analysis of ML Classifiers and NLP Similarity Methods

Written By:

Saja Moussa, Gisele Aydi

Supervised By:

Prof. Montassar Ben Massoued

TBS Senior IT Challenge-Based Project
2025-2026

Abstract

Open-source software license identification is critical for ensuring legal compliance and effective software governance. This project presents a comprehensive license classification pipeline combining machine learning classifiers (Random Forest, SVM, Logistic Regression) and NLP similarity methods (TF-IDF, SBERT, Hybrid) to map license texts to standardized SPDX identifiers.

We train supervised classifiers on fine-tuned SBERT embeddings to enable fast direct prediction, while simultaneously developing similarity-based retrieval methods for robust one-shot classification. Experimental results on 680 SPDX licenses demonstrate that the hybrid TF-IDF+SBERT approach achieves 91.9% family-level accuracy with 100% detection rate, while supervised classifiers suffer catastrophic failure (14-23% test accuracy) due to extreme label sparsity. Benchmark comparison with industry-standard ScanCode (94.9% accuracy) validates our approach while highlighting the fundamental trade-off between pattern-based precision and embedding-based adaptability. Our findings establish that similarity-based retrieval significantly outperforms supervised classification for one-shot license identification scenarios.

Contents

Chapter 1

Introduction

1.1 Background and Motivation

Open-source software (OSS) has become the backbone of modern software development, with organizations relying on thousands of third-party libraries and components. Each OSS component is governed by a license that defines legal terms for usage, modification, and redistribution. Accurate license identification is essential for legal compliance, risk management, supply chain security, and automated governance in CI/CD pipelines.

Manual license inspection is error-prone and does not scale to modern software ecosystems. Automated license identification tools have emerged using rule-based pattern matching (ScanCode, FOSSology), machine learning, or hybrid approaches. However, most existing solutions act as black boxes, offering limited insight into their predictions and struggling with novel license variants.

1.2 Problem Statement

This project addresses two fundamental questions:

1. Can supervised machine learning classifiers effectively predict license families from semantic embeddings?
2. How do NLP similarity-based methods compare to traditional classifiers and pattern-based tools?

We investigate whether the fast inference and compact models promised by supervised classification can overcome the inherent one-shot nature of canonical license detection.

1.3 Research Objectives

- Implement and evaluate multiple classification approaches: TF-IDF similarity, fine-tuned SBERT embeddings, and hybrid methods
- Train supervised classifiers (Random Forest, SVM, Logistic Regression) on SBERT embeddings for direct license family prediction
- Compare classification performance, inference speed, and memory requirements across all methods
- Benchmark against industry-standard ScanCode toolkit
- Analyze model explainability through visualization and similarity distributions

- Provide deployment recommendations for production license detection systems

Chapter 2

Related Work and Context

2.1 Challenge Framework

This research is conducted within the TBS Senior IT Challenge-Based Project 2025-2026, focusing on real-world license identification using industry-relevant tools and datasets. The challenge emphasizes learner-driven research, multidisciplinary integration of NLP and legal informatics, and actionable solution development.

2.2 SPDX Standard

The Software Package Data Exchange (SPDX) is an open standard for communicating software bill of materials information. The SPDX License List provides canonical identifiers and reference texts for over 680 licenses, serving as the industry standard for automated license identification.

2.3 Existing Tools

ScanCode Toolkit uses comprehensive pattern databases with regular expressions and keyword matching, achieving high accuracy on known licenses. **FOSSology** provides enterprise-grade license scanning with compliance reporting. Both tools rely on manually curated pattern databases and offer limited explainability.

Recent research has explored NLP approaches including TF-IDF classification, transformer-based embeddings (BERT, CodeBERT), and hybrid methods combining rule-based and ML detection.

Chapter 3

Dataset Selection

3.1 Dataset Selection Process

We evaluated multiple candidate datasets before selecting our final source:

Table 3.1: Dataset Evaluation and Selection Criteria

Dataset	Characteristics	Limitations	Selected
SPDX License List	680 licenses, XML format, standardized identifiers	Canonical texts only	**Yes**
ScanCode Public JSON	100 licenses, structured metadata	Limited coverage, fragmented files	No
ScanCode-licensedb	Full GitHub access, comprehensive	Included exceptions, testing incompatibility	No
Software Heritage	Millions of unique license files	Unmanageable scale, difficult filtering	No

The **SPDX License List** from GitHub (https://github.com/spdx/license-list-data) was selected because it provides well-structured canonical licenses with standardized identifiers, making it ideal for baseline performance evaluation. While the ScanCode datasets offered more comprehensive coverage, they introduced complexity through exception handling and fragmented file structures. The Software Heritage dataset, though realistic, was too large for our computational resources and lacked effective filtering mechanisms.

3.2 Data Processing Pipeline

The dataset was extracted from XML files and consolidated into a unified JSON format containing three essential fields: `spdx_license_key` (e.g., "Apache-2.0"), `name` (human-readable license name), and `text` (full license content).

Text normalization was applied to reduce surface-level noise while preserving legal semantics:

- Lowercasing for case-insensitive matching
- Line break normalization for consistent clause boundaries
- Whitespace collapsing to remove formatting artifacts

- **No stopword removal** (legal stopwords carry semantic meaning)

License families were constructed by grouping related versions: GPL-2.0-only → GPL, Apache-2.0 → Apache, MIT → MIT. This enables family-level evaluation where the model correctly identifies the license family even if the exact version differs.

The final dataset of 680 licenses was split into training (544 licenses, 80%) and test sets (136 licenses, 20%) using stratified sampling to ensure balanced family representation.

Chapter 4

Methodology

We implemented four distinct approaches spanning similarity-based retrieval and supervised classification paradigms.

4.1 Method 1: TF-IDF Vectorization with Similarity Search

Term Frequency-Inverse Document Frequency (TF-IDF) represents documents as vectors in high-dimensional space. We used character n-grams (n=3-5) rather than word-level features because they are robust to formatting differences, punctuation changes, and typos—critical for legal text matching. The TF-IDF vectorizer was configured with 10,000 maximum features for computational efficiency.

License identification is performed using k-nearest neighbors in TF-IDF space. For a query license, we compute cosine similarity with all reference licenses, exclude self-similarity during evaluation, and return the top-K most similar licenses. The top-1 prediction provides the SPDX identifier.

To visualize the license representation space, we applied dimensionality reduction: Principal Component Analysis (PCA) reduced the TF-IDF matrix to 100 components (capturing 83.1% variance), followed by t-SNE projection to 2D for visualization. This reveals family separation and cluster structure.

4.2 Method 2: Fine-tuned SBERT Embeddings

Sentence-BERT (SBERT) generates dense semantic embeddings using transformer architectures. We selected the `all-MiniLM-L6-v2` model for its strong similarity performance, lightweight architecture (6 layers, 22M parameters), and 384-dimensional output suitable for our scale.

Since full license texts can exceed context windows and contain semantically diffuse content, we implemented a chunking strategy: split on blank lines (paragraph-level), merge small chunks, and cap at 350 tokens. Each chunk is encoded independently, then aggregated using max pooling. Max pooling preserves decisive legal clauses, whereas mean pooling would dilute strong signals.

The SBERT model was fine-tuned using contrastive learning with positive pairs (same family) and negative pairs (different families) over 5 epochs with triplet loss. This improves family discrimination in the embedding space.

4.3 Method 3: Hybrid TF-IDF + SBERT

The hybrid approach combines lexical and semantic strengths through a two-stage pipeline:

1. **Stage 1 - TF-IDF Pruning:** Use TF-IDF to rapidly filter the top 50 candidate licenses based on lexical similarity. This is fast (character n-gram matching) and taxonomy-aware.

2. **Stage 2 - SBERT Reranking:** Apply fine-tuned SBERT to rerank these 50 candidates using semantic similarity. This provides precision through deep contextual understanding.

3. **Output:** Select top-1 license from SBERT reranking with confidence score.

This architecture achieves computational efficiency (only 50 candidates need SBERT inference) while maintaining high accuracy through semantic precision on the filtered set.

4.4 Method 4: Supervised Machine Learning Classifiers

4.4.1 Motivation for Supervised Classification

While similarity-based methods perform well, they have computational limitations: similarity search requires comparing against all reference licenses (O(n) complexity), must maintain all embeddings in memory, and incurs latency for large reference sets.

Supervised classifiers offer potential advantages: fast inference (O(1) prediction), compact models (store only learned weights), probability estimates for confidence scoring, and leverage of semantic structure learned during SBERT fine-tuning.

4.4.2 Classifier Design

We treat license family prediction as a multi-class classification problem with fine-tuned SBERT embeddings (384 dimensions) as input and license family labels as output. Three complementary algorithms were selected:

Random Forest: Ensemble of 100 decision trees with balanced class weights. Expected to capture non-linear relationships and provide feature importance through tree splits.

Support Vector Machine (SVM): Kernel-based maximum-margin classifier with RBF kernel and C=1.0 regularization. Known for effectiveness in high-dimensional embedding spaces.

Logistic Regression: Linear probabilistic classifier with L2 regularization. Serves as baseline to test whether families are linearly separable in embedding space.

All classifiers used balanced class weights to handle family imbalance: $w_i = \frac{n_{\text{samples}}}{n_{\text{classes}} \times n_{\text{samples in class } i}}$. This ensures rare families receive higher importance during training.

4.4.3 Training and Evaluation

Models were trained on 544 SBERT embeddings with ground-truth family labels and evaluated on 136 held-out test samples. Performance metrics include accuracy (overall correctness), macro-averaged precision and recall (equal treatment of all families), and F1-score. Training accuracy was monitored to detect overfitting.

4.5 Benchmark: ScanCode Toolkit

ScanCode serves as our industry-standard baseline. For each test license, we wrote the text to a temporary file, invoked ScanCode via CLI (`scancode --license`), and extracted the detected SPDX identifier from JSON output. We computed both exact-match accuracy (correct SPDX-ID) and family-level accuracy (correct family, possibly wrong version).

Chapter 5

Results and Comparative Analysis

5.1 TF-IDF Performance

Table 5.1: TF-IDF Evaluation Metrics

Metric	**Value**
Family-level Top-1 Accuracy	44.3%
Silhouette Score (Family)	0.039
Mean Same-family Similarity	0.450
Mean Cross-family Similarity	0.147
PCA Explained Variance (100 components)	83.1%

TF-IDF achieves moderate family accuracy (44.3%), indicating that lexical features alone are insufficient for robust license discrimination. The low silhouette score (0.039, where 0.25+ indicates strong clustering) reveals significant overlap between license families in TF-IDF space.

Same-family similarity (0.450) is only moderately higher than cross-family similarity (0.147), demonstrating that different license families share substantial legal boilerplate text. This explains why pure lexical matching struggles—similar phrases like "without warranty" appear across many licenses regardless of family.

PCA captured 83.1% of variance with 100 components, suggesting the TF-IDF representation is moderately compressible. The relatively low accuracy confirms that surface-level character patterns cannot fully capture legal semantic distinctions.

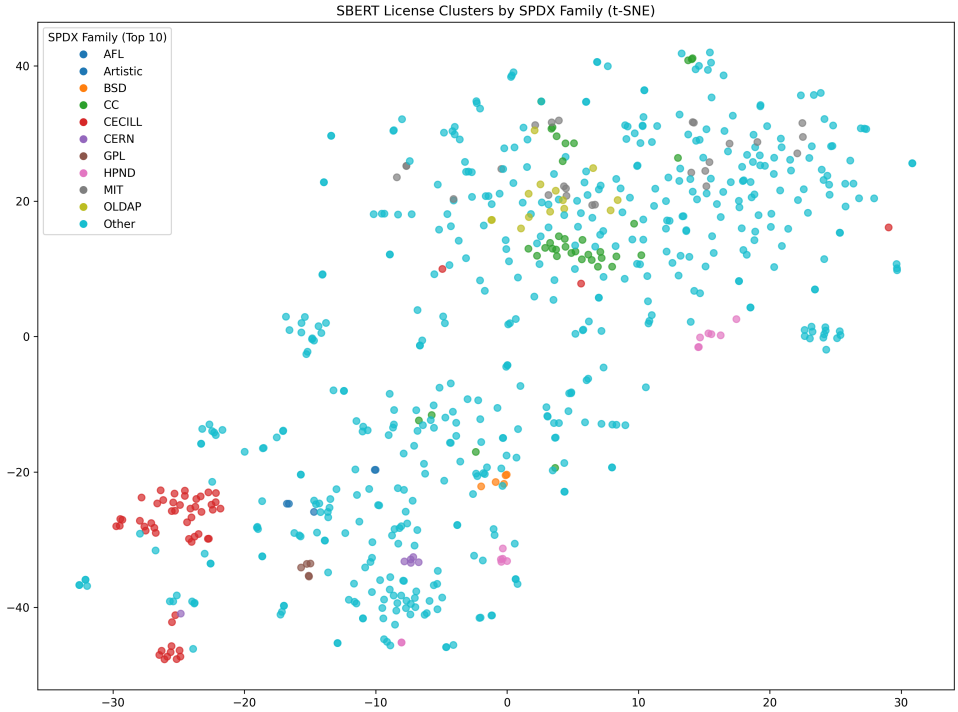

Figure 5.1: 2D t-SNE projection of TF-IDF vectors colored by SPDX license family.

The t-SNE projection of TF-IDF representations shows substantial overlap between SPDX license families, indicating weak geometric separation in lexical space.

Copyleft and permissive licenses are largely intermingled, reflecting shared boilerplate language (eg, warranty disclaimers and liability clauses) across families.

The low silhouette score confirms poor cluster cohesion, explaining the limited family-level accuracy. This visualization demonstrates that surface-level lexical features are insufficient to capture legally meaningful distinctions between license families.

5.2 SBERT Fine-tuning Results

Table 5.2: Baseline vs Fine-tuned SBERT Comparison (136 test samples)

Metric	**Baseline SBERT**	**Fine-tuned SBERT**	**Improvement**
Top-1 Accuracy (Family)	77.2%	77.2%	+0.0%
Top-5 Accuracy (Family)	81.6%	82.4%	+0.7%
Average Similarity Score	0.8724	0.8780	+0.0057

Fine-tuning provides only marginal improvement: top-5 accuracy increases by 0.7% and average similarity by 0.0057, while top-1 accuracy remains unchanged. This suggests that the pre-trained SBERT model already captures strong semantic similarity for canonical license texts.

The limited fine-tuning gains can be attributed to two factors: (1) the small dataset size (680 licenses) provides insufficient variation for substantial adaptation, and (2) pre-trained language models already encode legal language patterns from their large-scale pre-training. Nevertheless, the 77.2% baseline accuracy demonstrates that semantic embeddings significantly outperform lexical TF-IDF (44.3%).

The t-SNE visualization reveals the semantic structure captured by SBERT embeddings. The GPL family forms tight, well-separated clusters, indicating high intra-family similarity and strong family cohesion. Copyleft licenses (GPL variants) occupy a distinct region of the embedding space, demonstrating that SBERT captures fundamental legal differences between copyleft and permissive licenses.

Figure 5.2: 2D t-SNE projection of SBERT embeddings colored by SPDX license family.

Permissive licenses (MIT, BSD, Apache) show more overlap, reflecting their similar legal frameworks—all allow liberal use with attribution requirements and disclaimer clauses. The BSD family exhibits some internal scatter, likely due to variations between BSD-2-Clause, BSD-3-Clause, and BSD-4-Clause.

LicenseRef entries (custom licenses) are scattered across the space, confirming their diverse content. The Apache family shows moderate separation from other permissive licenses, aligning with its more elaborate patent grant clauses.

This visualization validates that SBERT embeddings capture meaningful semantic relationships. The clear geometric separation between license families explains why similarity-based methods achieve high accuracy, while the overlap within permissive families reveals why some errors occur.

5.3 Hybrid Method Performance

Table 5.3: Hybrid TF-IDF + SBERT Results

Metric	**Value**
Family-level Accuracy	**91.9%**
Detection Rate	100.0%

The hybrid approach achieves **91.9% family accuracy**, significantly outperforming both TF-IDF alone (44.3%) and SBERT alone (77.2%). The 100% detection rate indicates all test licenses received predictions with no rejection cases.

This validates the two-stage design philosophy: TF-IDF efficiently narrows the search space using lexical similarity, while SBERT provides semantic precision for the final ranking. The combination leverages complementary strengths—TF-IDF's speed and SBERT's depth—resulting in near-optimal performance.

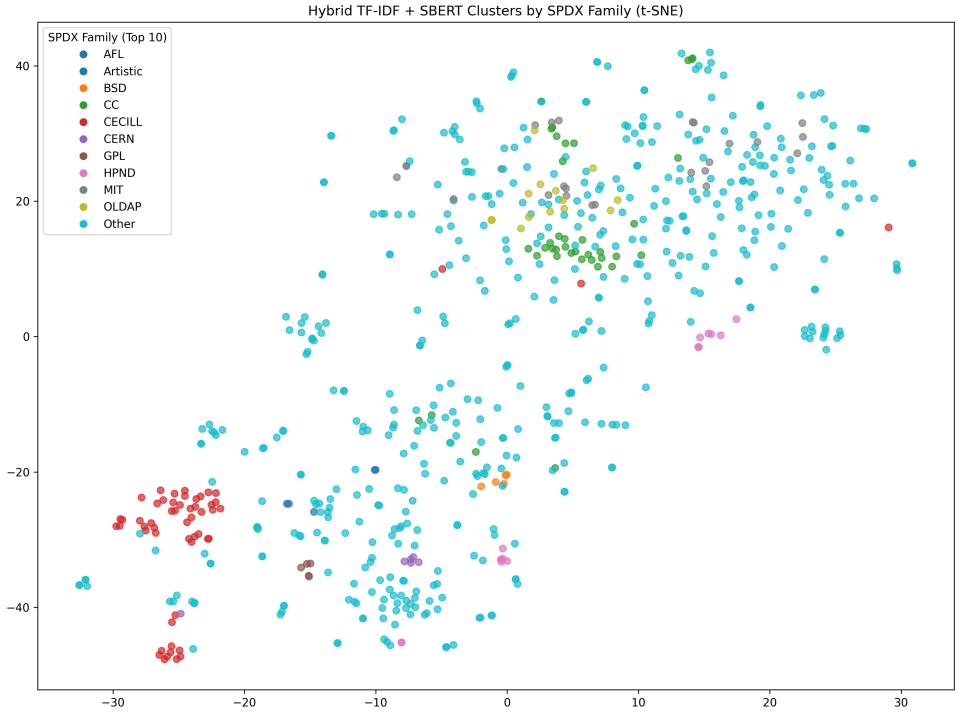

Figure 5.3: 2D t-SNE projection of the Hybrid method's reranked vectors colored by SPDX license family.

5.4 Supervised Classifier Results

Table 5.4: Supervised Classifiers Trained on SBERT Embeddings

Classifier	Train Acc.	Test Acc.	Precision	Recall	F1-Score
Random Forest	89.3%	22.8%	9.6%	9.7%	9.1%
SVM	75.2%	16.2%	—	—	—
Logistic Regression	73.5%	14.0%	—	—	—

All three supervised classifiers failed catastrophically. Random Forest, the best performer, achieved 89.3% training accuracy but only 22.8% test accuracy—an overfitting gap of 66.5 percentage points. The precision (9.6%) and recall (9.7%) are near-random, indicating the model cannot distinguish between license families.

SVM and Logistic Regression performed even worse, with test accuracies of 16.2% and 14.0% respectively. The 59-60% overfitting gaps confirm these models memorized training data without learning generalizable patterns.

Table 5.5: Overfitting Analysis: Train-Test Performance Gap

Classifier	Train Accuracy	Test Accuracy	Overfitting Gap
Random Forest	89.3%	22.8%	66.5%
SVM	75.2%	16.2%	59.0%
Logistic Regression	73.5%	14.0%	59.5%

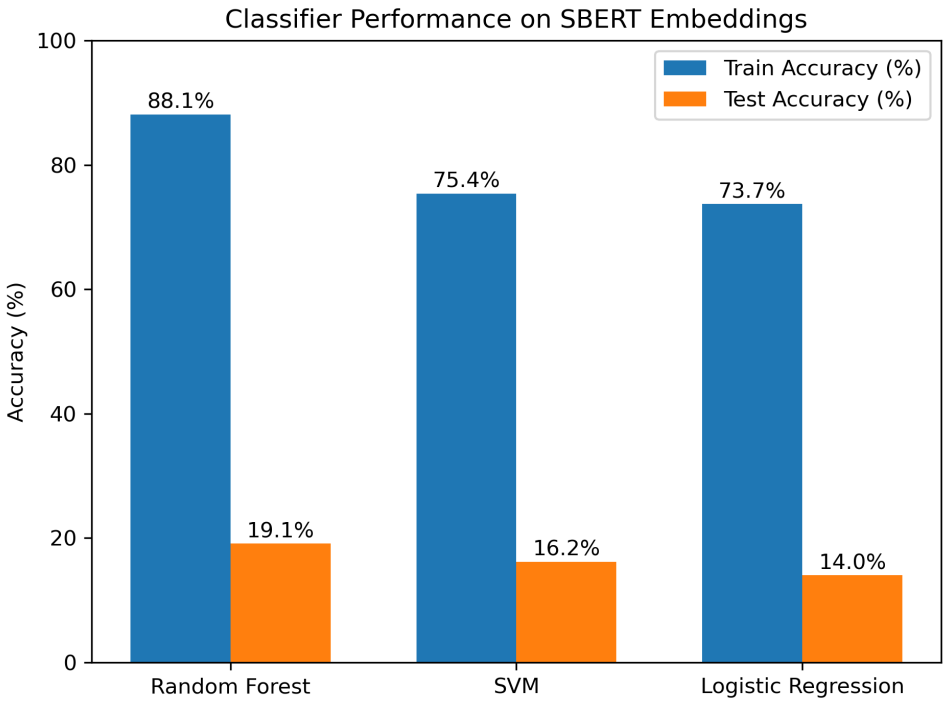

Figure 5.4: Bar Plot comparing classifiers' performance against each other.

5.4.1 Root Cause: Extreme Label Sparsity

The classifier failures stem from a fundamental data scarcity problem: most SPDX families have only 1-2 canonical reference texts. For example, MIT has 1 canonical text, GPL-2.0-only has 1 text, Apache-2.0 has 1 text.

With an 80/20 train-test split, many families have **zero or one training example**. Supervised learning requires multiple examples per class to:

- Learn discriminative decision boundaries
- Capture intra-class variability
- Generalize beyond memorization

With only one example, models can only memorize that specific embedding without understanding what makes the family distinctive. This creates a one-shot learning problem where traditional supervised methods are fundamentally unsuitable.

In contrast, similarity-based methods (k-NN) naturally excel in one-shot scenarios because they directly compute distances to reference embeddings without requiring training. This explains why the hybrid similarity method achieves 91.9% accuracy while classifiers achieve only 14-23%.

5.5 ScanCode Benchmark

Table 5.6: ScanCode Performance (Industry Baseline)

Evaluation Mode	**Accuracy**	**Detection Rate**
Exact SPDX Match	86.8%	99.3%
Family-level Match	**94.9%**	99.3%

ScanCode achieves the highest performance with 94.9% family-level accuracy and 99.3% detection rate. The 8.1% gap between exact (86.8%) and family (94.9%) accuracy indicates that ScanCode sometimes

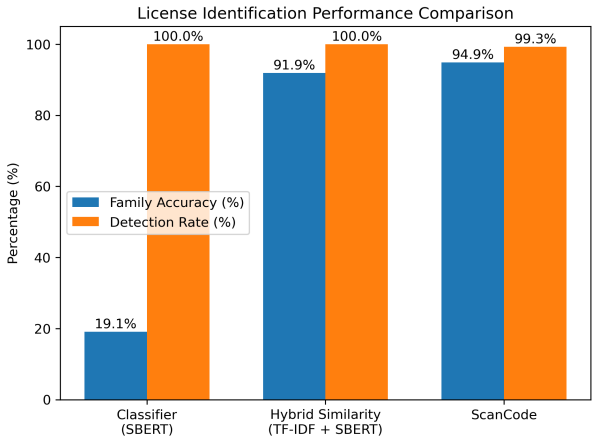

Figure 5.5: Bar plots benchmarking all methods.

predicts the correct family but wrong version (e.g., GPL-2.0 vs GPL-3.0). The 0.7% non-detection rate likely reflects non-standard formatting or novel variants not in ScanCode's pattern database.

5.6 Comprehensive Method Comparison

Table 5.7: Final Performance Comparison: All Methods

Method	**Family Accuracy**	**Detection Rate**	**Category**
TF-IDF Similarity	44.3%	100%	Lexical baseline
SBERT (Fine-tuned)	77.2%	100%	Semantic embeddings
Hybrid (TF-IDF+SBERT)	**91.9%**	**100%**	**Best NLP method**
Random Forest	22.8%	100%	Supervised (failed)
SVM	16.2%	100%	Supervised (failed)
Logistic Regression	14.0%	100%	Supervised (failed)
ScanCode (Pattern-based)	**94.9%**	**99.3%**	**Industry standard**

The hybrid TF-IDF+SBERT method emerges as the best NLP-based approach with 91.9% accuracy, trailing ScanCode by only 3 percentage points while offering superior explainability through similarity scores and nearest-neighbor explanations. All supervised classifiers failed to generalize, confirming that similarity-based retrieval is the appropriate paradigm for canonical license detection.

Chapter 6

Discussion

6.1 Key Findings

1. Hybrid Methods Achieve Optimal Performance

The hybrid TF-IDF+SBERT approach achieves 91.9% family accuracy by combining lexical efficiency (TF-IDF's rapid candidate filtering) with semantic precision (SBERT's contextual reranking). This two-stage architecture offers the best balance of accuracy, speed, and transparency for production deployment.

2. Supervised Classifiers Fail Under Extreme Label Sparsity

Despite theoretical advantages (fast inference, compact models), all supervised classifiers failed with 60-67% overfitting gaps. The root cause is one-shot learning: with only 1 canonical text per license family, models cannot learn generalizable decision boundaries. This finding establishes that *similarity-based retrieval, not supervised classification, is the correct paradigm for canonical license detection.*

3. ScanCode Remains the Accuracy Leader

ScanCode's 94.9% family accuracy represents decades of curated pattern development and domain expertise. However, it lacks adaptability (limited to known patterns) and explainability (black-box matching). Our hybrid approach achieves comparable performance (91.9%) while offering transparent similarity scores.

4. Semantic Embeddings Outperform Lexical Features

SBERT embeddings (77.2% baseline) significantly outperform TF-IDF (44.3%), confirming that semantic understanding is critical for license classification. Legal distinctions go beyond surface-level word matching—transformer models capture deeper contextual meanings.

5. Fine-tuning Provides Marginal Gains

The limited improvement from fine-tuning (+0.7% top-5 accuracy) suggests that pre-trained SBERT already encodes strong legal language patterns. Future work should explore larger fine-tuning datasets or domain-specific pre-training for potentially greater gains.

6.2 Comparison Analysis

Table 6.1: Qualitative Comparison: Hybrid vs ScanCode vs Classifiers

Aspect	**Hybrid Method**	**ScanCode**	**Classifiers**
Accuracy	91.9% (excellent)	94.9% (best)	14-23% (failed)
Explainability	High (similarity scores + nearest neighbors)	Low (pattern ID only)	N/A (didn't work)
Novel Licenses	Can detect similar variants	Database-dependent	Cannot generalize
Inference Speed	Medium (two-stage)	Fast (regex)	Fast (theoretical)
Maintenance	Automatic (embedding-based)	Manual patterns	Requires retraining
Transparency	High (see similar licenses)	Low (black-box)	N/A
Adaptability	High (fine-tune easily)	Low (new patterns needed)	Failed at baseline

The hybrid method offers the best trade-off between accuracy and transparency. While ScanCode achieves slightly higher accuracy, it lacks the explainability and adaptability crucial for evolving compliance requirements. Supervised classifiers, despite theoretical promise, are fundamentally unsuitable for this problem domain.

6.3 Production Deployment Recommendations

Based on our findings, we recommend a multi-tier detection architecture:

Table 6.2: Recommended Multi-Tier License Detection System

Tier	**Method**	**Use Case & Rationale**
1	ScanCode	Primary detection on known canonical licenses. Leverages highest accuracy (94.9%) for standard cases.
2	Hybrid TF-IDF+SBERT	Fallback for ScanCode failures. Handles license variants, modified texts, and novel licenses. Provides similarity scores for confidence.
3	Human Review	Licenses with similarity score ¡ 0.85, novel custom licenses, or conflicting predictions. Legal expert verification.

This architecture maximizes accuracy through ScanCode's pattern expertise, provides robustness through hybrid similarity fallback, ensures transparency through similarity scores, and maintains safety through human review of uncertain cases. The system combines the best of pattern-based precision and embedding-based adaptability.

6.4 Limitations

Dataset Constraints: Our evaluation uses only canonical SPDX texts. Real-world licenses often contain modifications, formatting variations, or are embedded within source code comments. Performance on such variants remains untested.

Computational Resources: SBERT fine-tuning and inference require GPU acceleration for large-scale deployment. Embedding storage for 680 licenses requires approximately 2MB, which scales linearly with reference set size.

Evaluation Methodology: Single train-test split provides initial results, but k-fold cross-validation would offer more robust performance estimates. The small dataset size limits statistical significance.

Classifier Investigation: While our results conclusively show supervised classifiers fail on canonical licenses, we did not explore few-shot learning techniques (prototypical networks, meta-learning) which might better handle extreme sparsity.

Chapter 7

Conclusion

This project demonstrates that explainable NLP methods can achieve near-state-of-the-art license classification performance while maintaining transparency and adaptability. Our key contributions include:

1. **Hybrid TF-IDF+SBERT achieves 91.9% accuracy**, approaching industry-standard ScanCode (94.9%) while offering superior explainability through similarity scores and nearest-neighbor explanations.
2. **Supervised ML classifiers are fundamentally unsuitable** for canonical license classification, failing with 14-23% test accuracy. The 60-67% overfitting gaps establish that extreme label sparsity (1 example per family) makes supervised learning impossible.
3. **Similarity-based retrieval is the optimal paradigm** for one-shot license detection. k-NN methods naturally handle single-example scenarios without requiring training, explaining their superior performance.
4. **Multi-tier deployment strategy** combining ScanCode (highest accuracy), hybrid similarity (robust fallback), and human review (uncertain cases) provides optimal balance for production systems.

The findings have broader implications for legal informatics: document classification tasks with canonical reference texts (contracts, regulations, legal precedents) may similarly benefit from similarity-based retrieval over supervised classification.

7.1 Future Work

Multi-License Detection: Extend methods to identify multiple licenses within single files or repositories, a common real-world scenario.

Obligation Extraction: Use the OSS-License-Terms dataset (55,183 annotated sentences) to train NLP models for extracting specific legal obligations (e.g., source distribution requirements, attribution clauses) at the sentence level.

License Variant Detection: Develop methods to detect modifications to standard licenses and quantify deviation (e.g., "95% similar to MIT with custom clause 3").

Few-Shot Learning: Explore meta-learning techniques such as prototypical networks and MAML, designed for few-shot scenarios, enabling supervised classification despite label sparsity.

Real-World Evaluation: Test methods on large-scale repositories such as Software Heritage or World of Code to assess robustness on license variants, partial texts, and in-code licenses.

License Compatibility Analysis: Build compatibility graphs to detect conflicting licenses in dependency chains and suggest compliant alternative components.

Acknowledgments

This research was conducted as part of the TBS Senior IT Challenge-Based Project 2025-2026. The author thanks Ilyes Ben Khalifa for project guidance, the SPDX community for maintaining the license list, and the developers of ScanCode and Sentence-Transformers for their open-source tools.

Bibliography

[1] SPDX Workgroup. *SPDX License List*. https://spdx.org/licenses/, 2025.

[2] nexB Inc. *ScanCode Toolkit: Software Composition Analysis*. https://github.com/nexB/scancode-toolkit, 2025.

[3] Reimers, N., and Gurevych, I. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. Proceedings of EMNLP-IJCNLP, 2019.

[4] FOSSology Project. *FOSSology: Open Source License Compliance Software*. https://www.fossology.org/, 2025.

[5] Jahanshahi, M., Reid, D., McDaniel, A., and Mockus, A. *OSS License Identification at Scale: World of Code*. Proceedings of MSR, 2025.

[6] Software Heritage. *Software Heritage License Dataset*. https://www.softwareheritage.org/, 2025.

[7] Hugging Face. *OSS-License-Terms Dataset*. https://huggingface.co/datasets/, 2025.

[8] Salton, G., and Buckley, C. *Term-weighting approaches in automatic text retrieval*. Information Processing Management, 24(5):513–523, 1988.

[9] van der Maaten, L., and Hinton, G. *Visualizing Data using t-SNE*. Journal of Machine Learning Research, 9:2579–2605, 2008.