A Comparative Study of Automated Approaches for Detecting Subjectivity and Unverifiability in Natural Language Software Requirements

Muhammad Arsalan Iltaf, Aamer Nadeem
Capital University of Science and Technology, Islamabad.
*__Correspondence__: arsalaniltaf@hotmail.com

Citation | Iltaf. M. A, Nadeem. A, "A Comparative Study of Automated Approaches for Detecting Subjectivity and Unverifiability in Natural Language Software Requirements", IJIST, Special Issue pp 75-86, April 2026
Received | March 25, 2026 **Revised |** April 20, 2026 **Accepted |** April 25, 2026 **Published |** April 28, 2026.

Natural language software requirements are prone to subjectivity and unverifiability, which reduces clarity and testability. This study provides a controlled comparative evaluation of four automated detection approaches: rule-based lexicons, classical machine-learning models with TF-IDF features, a fine-tuned DistilBERT transformer, and a feature-level hybrid model that concatenates rule-based linguistic indicators with DistilBERT contextual embeddings on the same manually annotated dataset of 985 requirements, including 259 subjective and 726 objective items, and 165 unverifiable and 820 verifiable items. All models used identical preprocessing, an 80:20 stratified train-test split, and the same evaluation metrics. The rule-based approach achieved a precision of 0.67 and a recall of 0.04 for subjectivity and zero recall for unverifiability. Classical ML models reached F1-scores ranging from 0.47 to 0.58 (positive class). DistilBERT obtained weighted F1-scores of 0.74 (subjectivity) and 0.86 (unverifiability). The feature-level hybrid model improved subjectivity detection to a weighted F1-score of 0.79 while matching DistilBERT at 0.86 for unverifiability. These results demonstrate that combining explicit linguistic cues with contextual embeddings improves detection performance under class imbalance.

Keywords: Requirement Smells; Subjectivity Detection; Unverifiability Detection; Transformer-Based Models; Hybrid Machine-Learning; DistilBERT

Introduction:

The most widely used form of specifying software requirements is through natural language (NL) because of its flexibility, expressiveness, and readability to all stakeholders who may have varying levels of knowledge of the technical requirements [1]. Although these benefits are well recognized, NL requirements are susceptible to quality issues that may lead to misunderstandings, rework, high costs of development, and system failures. Empirical research in requirements engineering has shown that defects introduced during the requirements phase are among the most costly to rectify, and studies on requirements quality and the prevalence of requirement smells further support this observation [2][3][4]. As a result, the quality of natural language requirements remains a significant research challenge [5]. In order to cope with this problem, research in requirements engineering focuses on the detection and prevention of linguistic quality problems, which are commonly referred to as requirement smells. Requirement smells are not literal errors but indicators of likely flaws in a specification to likely flaws in a specification; they point out sections of a specification that could contravene the desired quality attributes like clarity, objectivity, and testability [6]. Ambiguity in requirements has been studied as a fundamental linguistic phenomenon with multiple interpretations rather than a single concrete defect [7]. Among the various requirement smells discussed in the literature, subjectivity and unverifiability are particularly critical due to their impact on requirement clarity and testability. Unverifiable requirements, on the other hand, do not have any measurable acceptance criteria and, therefore, cannot be easily or objectively validated and tested. Both smells contradict international standards (e.g., ISO/IEC/IEEE 29148), which require software requirements to be objective, measurable, and testable [8]. Initial approaches for detecting subjectivity and unverifiability were primarily based on manual inspection and rule-based linguistic checks. These methods rely on lexical patterns and linguistic heuristics created based on the knowledge and standards of the experienced linguists [7][29]. Rule-based techniques are interpretable but rely on surface-level linguistic cues and typically exhibit low recall, as they fail to capture implicit or contextualized expressions. Researchers have addressed these limitations by applying classical machine-learning methods based on bag-of-words and TF–IDF representations [10][11][12]. Although these methods improve scalability and automation, they rely on surface-level lexical features, which limit their ability to capture deeper semantic relationships. Recent progress in deep learning and natural language processing has brought on board the transformer-based language models as an effective alternative to the analysis of text. Transformer-based architectures such as BERT and its lightweight variants have demonstrated strong performance in capturing contextual and semantic information across a wide range of NLP tasks [13], including recent applications for detecting semantic data smells using transformer-based models [14]. Transformer-based models used in requirements engineering provide the capability to sufficiently detect implicit subjectivity and unverifiability when explicit handcrafted rules are not readily applicable. Nonetheless, purely data-driven models can ignore explicit normative information that is specified in requirements standards and can be hard to make sense of, which is a crucial factor in software areas with high safety and quality concerns.

Recent advances in natural language processing have significantly influenced requirements engineering research [15][16][17], particularly in the automated detection of requirement smells. Transformer-based models such as BERT and its variants, built upon the transformer architecture [18][19], have demonstrated strong capability in capturing contextual semantics for software engineering tasks [20][21]. Recent studies (2021–2025) have applied deep learning approaches, including contextual embedding techniques, domain-adaptive models, and multi-label classification frameworks, to improve the detection of implicit linguistic defects in requirements [22][23][24][25]. In addition, hybrid approaches that combine symbolic linguistic knowledge with data-driven models have been explored to enhance both

detection performance and interpretability [26][27][28][29]. These developments highlight the importance of context-aware and integrated representations for identifying subtle requirement quality issues. Despite recent advances in automated detection of requirement smells using rule-based, machine-learning, and transformer-based approaches, several limitations remain. Existing studies often focus on individual detection paradigms in isolation or target specific types of requirement smells without providing a unified comparison. Furthermore, many approaches rely either on surface-level linguistic features or purely contextual representations, without systematically integrating both within a single learning framework. In addition, limited work has evaluated these approaches under a unified experimental setup using the same dataset, preprocessing pipeline, and evaluation metrics, particularly for subjectivity and unverifiability detection. Therefore, a clear gap exists in the absence of controlled comparative evaluation of rule-based, classical machine-learning, transformer-based, and hybrid approaches for detecting subjectivity and unverifiability under identical experimental conditions.

Research Objectives: The main objectives of this study are as follows:

A controlled comparative evaluation of rule-based, classical machine-learning, transformer-based, and feature-level hybrid approaches for detecting subjectivity and unverifiability in natural language software requirements using a unified dataset and evaluation protocol.

The incorporation and evaluation of a feature-level hybrid approach that integrates rule-based linguistic indicators with transformer-based contextual embeddings within the comparative experimental framework.

An empirical analysis of detection performance under class imbalance, using precision, recall, and F1-score metrics, provides insight into the behavior of different paradigms across both subjectivity and unverifiability tasks.

A reproducible experimental setup that ensures fair comparison across methods by maintaining consistent preprocessing, dataset, and evaluation conditions.

Research Questions: In an attempt to operationalize such objectives, the research questions are as follows:

RQ1: What is the difference between rule-based, classical machine-learning, and transformer-based methods in finding subjectivity and unverifiability in natural language software requirements?

RQ2: What is the effect of feature-level hybridization on detection behavior when there is an imbalance in classes? The primary contribution of this work is the evaluation of a feature-level hybrid model integrating linguistic indicators and contextual embeddings, along with a comprehensive comparative analysis of multiple detection approaches for identifying requirement smells.

To address the defined research questions, a controlled experimental design is adopted in which all detection approaches are evaluated under identical conditions using the same dataset, preprocessing pipeline, and train–test split. Research Question 1 (RQ1) is addressed by comparing the performance of rule-based, classical machine-learning, transformer-based, and feature-level hybrid approaches using precision, recall, and F1-score metrics. These metrics provide a quantitative basis for analyzing differences in detection capability across models.

Research Question 2 (RQ2) is addressed by examining the performance of the feature-level hybrid model in comparison to other approaches under class imbalance conditions. The use of weighted F1-score, along with class-wise precision and recall, enables evaluation of detection behavior across imbalanced classes. This experimental setup ensures that each research question is directly linked to measurable evaluation criteria.

Materials and Methods: This study evaluates multiple detection paradigms under a controlled experimental setup using a unified dataset and evaluation framework.

Dataset Description:

The dataset used in this study consists of 985 natural language software requirements, each annotated for two requirement smells: subjectivity and unverifiability. The dataset includes 259 subjective and 726 objective requirements, as well as 165 unverifiable and 820 verifiable requirements, indicating class imbalance in both classification tasks. The dataset is a curated collection compiled from publicly available requirement examples and datasets commonly used in requirements engineering and machine-learning research.

Some requirements include source indicators (e.g., "ERTMS"), suggesting their origin from real-world specification documents [3][30]. All requirements were manually annotated following established definitions of requirement smells in the literature. Subjectivity refers to opinion-based or qualitative expressions (e.g., "user-friendly," "efficient"), whereas unverifiability refers to requirements that lack measurable or testable criteria, such as missing quantitative thresholds or acceptance conditions. To ensure consistency, predefined annotation guidelines based on these linguistic characteristics were applied uniformly across the dataset.

Although formal inter-annotator agreement metrics are not reported, the annotation process follows standardized definitions widely adopted in prior work. This makes the dataset suitable for controlled comparative evaluation of automated approaches under consistent experimental conditions.

Proposed Methodology: This section outlines the general approach for detecting subjectivity and unverifiability in natural language software requirements. To allow a fair comparison of the various detection paradigms, such as rule-based, classical machine-learning, transformer-based, and hybrid methodologies, the proposed methodology adheres to a controlled and systematic experimental design. The overall workflow of the proposed approach is illustrated in Figure 1.

The process begins with dataset preparation and preprocessing, where textual requirements are normalized to ensure consistent input across all models. Following preprocessing, four detection paradigms are applied in parallel: rule-based detection, classical machine-learning models using TF–IDF features, a transformer-based model (DistilBERT), and a feature-level hybrid model. In the hybrid approach, rule-based linguistic indicators are combined with contextual embeddings generated by the transformer model to form a unified feature representation. All models are trained and evaluated using the same dataset, preprocessing pipeline, and train–test split to ensure a fair comparison. The outputs of each approach are then evaluated using precision, recall, and F1-score metrics to analyze detection performance.

Preprocessing: Following dataset preparation, all requirements are processed to ensure consistent input across all models. Preprocessing includes text lowercasing, removal of extraneous whitespaces, and basic text normalization. Aggressive linguistic preprocessing techniques such as stemming and lemmatization are intentionally not applied. This decision is motivated by the use of transformer-based models, which rely on contextual and subword tokenization mechanisms that preserve semantic and syntactic information in the original text. Applying stemming or lemmatization may alter word forms and reduce contextual richness, potentially affecting the performance of such models. Additionally, maintaining the original lexical structure ensures consistency across all evaluated approaches, enabling a fair comparison between rule-based, classical machine-learning, and transformer-based methods. This design choice supports the controlled experimental setup and allows performance differences to be attributed to the models rather than variations in preprocessing.

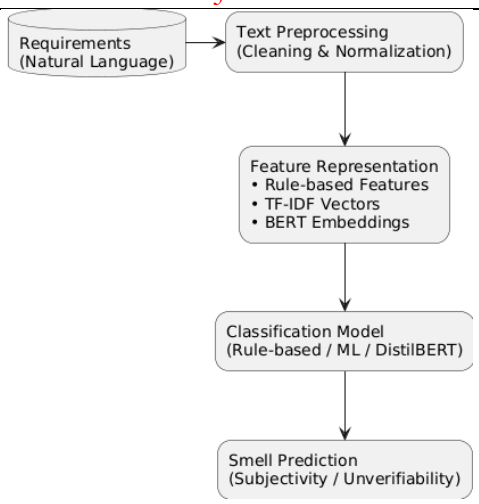

Figure 1. Overall Experimental Framework

Rule-Based Detection: The rule-based approach is used as a benchmark for the identification of explicit linguistic signs of subjectivity and unverifiability. The subjectivity is determined on the basis of pre-defined lists of adjectives that are expressed in opinion and words of ambiguity that are usually used in the requirements engineering literature, including easy, user-friendly, and efficient. To formalize the rule-based detection process, a lexicon-driven scoring mechanism was employed. A predefined dictionary of subjective and unverifiable terms (e.g., user-friendly, efficient, fast, adequate) was constructed based on prior literature and manual inspection of the dataset. Each requirement R_i is represented as a sequence of tokens$\{w_1, w_2, \ldots, w_n\}$. A subjectivity score is computed as:

$$S(R_i) = \frac{\sum_{j=1}^{n} \mathbb{I}(w_j \in L_s)}{n}$$

Where L_s denotes the lexicon of subjective terms, and $\mathbb{I}$ is an indicator function that returns 1 if the token belongs to the lexicon and 0 otherwise.

Similarly, an unverifiability score is defined as:

$$U(R_i) = \frac{\sum_{j=1}^{n} \mathbb{I}(w_j \in L_u)}{n}$$

Where L_u represents the set of unverifiable terms.

A requirement is classified as subjective or unverifiable if the corresponding score exceeds a predefined thresholdθ:

$$\text{Label}(R_i) = \begin{cases} 1, & \text{if } S(R_i) > \theta_s \text{ or } U(R_i) > \theta_u \\ 0, & \text{otherwise} \end{cases}$$

In this study, threshold values θ_s and θ_u were empirically determined through preliminary experiments on a validation subset of the dataset. This formulation provides a consistent and reproducible mechanism for identifying vague or unverifiable expressions and represents a generalized abstraction of the rule-based detection process implemented in this study. Unverifiability is detected by identifying vague or qualitative terms that appear without associated quantitative constraints (e.g., "fast" or "sufficient"). The existence of numerical values, units, or measurable constraints is identified with the use of regular expressions. A requirement that contains vague expressions but lacks measurable criteria is classified as unverifiable. Although such a strategy is clear and can be read, it is conservative and limited to explicit surface-level cues, resulting in low recall.

Model Selection Rationale: The reason why the methods of detection were chosen in this research is that it covers a representative range of paradigms that are predominantly employed when researching requirements engineering. The rule-based approach falls under the

interpretable baseline, which is based on known heuristics in the linguistics domain. TF -IDF representations using classical machine-learning models are statistical baselines and are popularly used in text-detection tasks. DistilBERT is chosen as a transformer-based model because it simultaneously allows capturing contextual semantics and has a lower computational cost than larger transformer models like BERT. Lastly, the feature-level hybrid model aims to investigate how the idea of combining symbolic linguistic knowledge and contextual embeddings within a unified learning framework.

Classical Machine-Learning Models: To establish statistical baselines within the comparative framework, classical machine-learning models are implemented using TF–IDF representations of the requirements text. In this approach, each requirement is transformed into a high-dimensional feature vector using Term Frequency-Inverse Document Frequency (TF–IDF), capturing the relative importance of terms within the corpus. Three supervised classifiers are considered: Logistic Regression, Linear Support Vector Machine (SVM), and Random Forest. These models are selected due to their established effectiveness in text classification tasks and their suitability as baseline approaches in requirements engineering research. Each model is trained separately for subjectivity detection and unverifiability detection to ensure task-specific evaluation. To address class imbalance in the dataset, class weighting is applied during training, assigning higher importance to minority classes. This enables more balanced learning and prevents bias toward the majority class. Within the controlled experimental setup, these models provide a statistical baseline for comparison with rule-based, transformer-based, and feature-level hybrid approaches. However, their reliance on surface-level lexical features limits their ability to capture deeper semantic and contextual relationships, which are often necessary for identifying implicit subjectivity and unverifiability in natural language requirements.

Transformer-Based Model: A transformer-based model is employed to capture contextual and semantic information using DistilBERT, a lightweight version of BERT that provides a balance between performance and computational efficiency. DistilBERT is trained using a binary classification setup for each task, where subjectivity detection and unverifiability detection are modeled independently. This ensures consistency with classical machine-learning models and maintains a controlled comparison across all approaches under identical experimental conditions. A lightweight transformer architecture is practically applicable and lowers training and inference expenses compared to larger transformer models. All the requirements are tokenized with the DistilBERT tokenizer and truncated or padded to a fixed length. The contextual representation of the [CLS] token is passed to a classification layer to generate predictions. The model is trained using binary cross-entropy loss with logits, and predictions are obtained using a threshold-based decision rule.

Feature-Level Hybrid Model: The architecture of the feature-level hybrid model is illustrated in Figure 2, showing the integration of rule-based features with transformer-based contextual embeddings. The hybrid model combines rule-based linguistic indicators with contextual embeddings based on transformers at the feature level. Rule-based features that describe the existence of subjective words, expressions that cannot be verified, and quantitative constraints are obtained for each requirement. These symbolic features are concatenated with the contextual embedding that is generated by DistilBERT. The combined feature vector is then fed to a classification head in the classification layer. In contrast to decision-level fusions, such feature-level fusion enables the model to learn the relative weighting of symbolic and contextual information during training. The design integrates rule-based interpretability with transformer-based semantic expressiveness, to create a more robust detector even in the face of class imbalance.

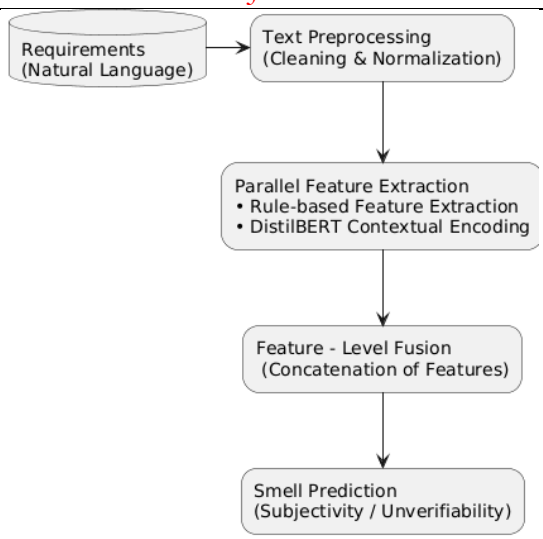

Figure 2. Feature-Level Hybrid Model

Experimental Setup and Hyperparameter Configuration: To ensure reproducibility and fair comparison, all models are evaluated under a controlled experimental setup using the same dataset, preprocessing pipeline, and stratified train-test split. The dataset is divided into training and testing sets using an 80:20 ratio with a fixed random state of 42, preserving class distribution for both subjectivity and unverifiability labels.

For classical machine-learning models, TF–IDF vectorization is applied using both unigram and bigram features (n-gram range $= (1,2)$), with a maximum of 5000 features and English stop-word removal. Logistic Regression is configured with L2 regularization, a maximum of 1000 iterations, and the 'liblinear' solver. The Linear Support Vector Machine (SVM) is implemented using a linear kernel with class balancing. The Random Forest classifier is configured with 200 decision trees and class weighting to address class imbalance. All classical machine-learning models are implemented using a binary classification setup for each task, where subjectivity detection and unverifiability detection are modeled independently. This ensures consistency with the transformer-based and feature-level hybrid approaches and supports a controlled comparison across all models.

The DistilBERT model is fine-tuned using a maximum sequence length of 128 tokens, with input sequences padded or truncated accordingly. The model is trained for 4 epochs using a learning rate of 2×10^{-5}, a batch size of 16, and the AdamW optimizer. Binary cross-entropy loss with logits is used for binary classification in each task, and a threshold of 0.5 is applied for prediction.

For the feature-level hybrid model, contextual embeddings from DistilBERT are concatenated with rule-based linguistic features, including indicators for subjective terms, unverifiable expressions, and the presence of numerical constraints. The combined feature representation is passed through a fully connected classification layer with a dropout rate of 0.3. The hybrid model is trained using the same hyperparameter settings as the transformer-based model to ensure a fair comparison.

The rule-based baseline is implemented using predefined lexicons for subjective and unverifiable terms, along with regular expression patterns to detect numerical constraints. Predictions are generated based on the presence of these linguistic indicators. All models are evaluated under identical experimental conditions without cross-validation, ensuring that performance differences are attributable to model characteristics rather than variations in training configuration.

Due to class imbalance in both subjectivity and unverifiability tasks, the weighted F1-score is considered the primary evaluation metric for performance comparison. F1-score for

the positive (smell) class is reported as a complementary measure to assess detection capability for minority classes.

Results and Discussion:

This section presents the experimental results of the rule-based baseline, classical machine-learning models, transformer-based model, and the proposed feature-level hybrid model. All models were evaluated under identical conditions to ensure a fair comparison.

Figure 3 illustrates the comparative performance of all evaluated models across subjectivity and unverifiability detection tasks, highlighting the improvements achieved by the transformer-based and hybrid approaches.

Table 1 summarizes the performance of all evaluated models for subjectivity and unverifiability detection under a controlled experimental setup.

The rule-based baseline achieves a precision of 0.67 and a recall of 0.04 for subjectivity detection, and a recall of 0.00 for unverifiability detection, indicating very limited capability in detecting minority (smell) classes.

Classical machine-learning models using TF–IDF representations achieve noticeably higher F1-scores than the rule-based baseline. For subjectivity detection, these models attain F1-scores ranging from 0.47 to 0.58 (compared to 0.07 for the rule-based approach).

Table 1. Results

Model	Smell	Class	Precision	Recall	F1-score	Weighted F1-score
Feature-Level Hybrid	Subjectivity	Objective 0	0.84	0.89	0.86	0.79
		Subjective 1	0.63	0.52	0.57	
	Unverifiability	Verifiable 0	0.89	0.96	0.93	0.86
		Unverifiable 1	0.70	0.42	0.53	
DistilBERT-only	Subjectivity	Objective 0	0.84	0.79	0.81	0.74
		Subjective 1	0.50	0.60	0.54	
	Unverifiability	Verifiable 0	0.89	0.96	0.93	0.86
		Unverifiable 1	0.70	0.42	0.53	
TF–IDF + Linear SVM	Subjectivity	Objective 0	0.81	0.89	0.85	0.75
		Subjective 1	0.57	0.40	0.47	
	Unverifiability	Verifiable 0	0.90	0.95	0.92	0.85
		Unverifiable 1	0.63	0.46	0.53	
TF–IDF + Logistic Regression	Subjectivity	Objective 0	0.81	0.82	0.82	0.72
		Subjective 1	0.48	0.46	0.47	
	Unverifiability	Verifiable 0	0.91	0.95	0.93	0.87
		Unverifiable 1	0.65	0.52	0.58	
TF–IDF + Random Forest	Subjectivity	Objective 0	0.76	0.99	0.86	0.68
		Subjective 1	0.75	0.12	0.20	
	Unverifiability	Verifiable 0	0.88	0.98	0.93	0.84
		Unverifiable 1	0.77	0.30	0.44	
Rule-Based Baseline	Subjectivity	Objective 0	0.74	0.99	0.85	0.65
		Subjective 1	0.67	0.04	0.07	
	Unverifiability	Verifiable 0	0.83	0.99	0.91	0.75
		Unverifiable 1	0.00	0.00	0.00	

For unverifiability detection, Logistic Regression achieves the highest F1-score of 0.58 (compared to 0.00 for the rule-based baseline). In terms of weighted F1-score, Linear SVM (0.75 for subjectivity) and Logistic Regression (0.87 for unverifiability) outperform the Random Forest model (0.68 and 0.84, respectively). The transformer-based DistilBERT model achieves weighted F1-scores of 0.74 for subjectivity and 0.86 for unverifiability, performing

comparably to the best classical machine-learning models (e.g., 0.87 weighted F1 by Logistic Regression for unverifiability).

The feature-level hybrid model further improves performance, achieving a higher weighted F1-score of 0.79 (versus 0.74 for DistilBERT) for subjectivity detection while maintaining the same weighted F1-score of 0.86 for unverifiability. This suggests improved robustness under class imbalance conditions. The confusion matrix shows that the rule-based approach yields a high number of false negatives, particularly for subjectivity and unverifiability.

Transformer-based and hybrid models substantially reduce false negatives, although some misclassification remains for requirements containing vague qualifiers without measurable criteria. The experimental findings directly address the defined research questions.

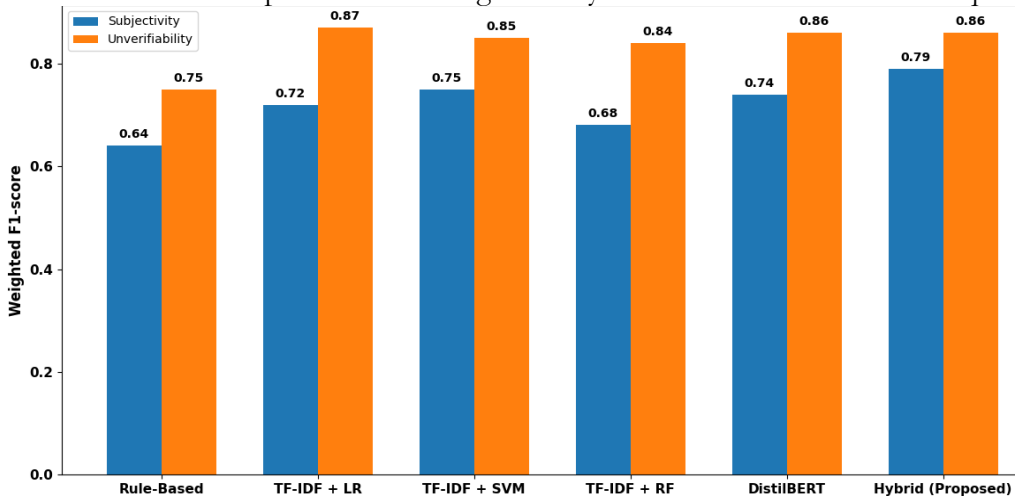

Figure 3. Comparison of Results.

Regarding RQ1, the results demonstrate clear performance differences across paradigms. Classical machine-learning models achieve F1-scores ranging from 0.47 to 0.58 for subjectivity (versus 0.07 for rule-based), while the transformer-based model reaches a weighted F1-score of 0.74. For unverifiability, classical models achieve F1-scores up to 0.58, whereas the transformer-based model achieves 0.86 (versus 0.00 for rule-based).

Regarding RQ2, the feature-level hybrid model improves subjectivity detection under class imbalance, achieving a weighted F1-score of 0.79 compared to 0.74 for the transformer-based model, while maintaining comparable performance for unverifiability (0.86).

Discussion:

The experimental results demonstrate clear performance differences across detection paradigms under identical experimental conditions. The rule-based approach, which relies on predefined lexical patterns, shows limited detection capability, particularly for the minority classes, as reflected by very low recall values. This confirms that rule-based methods are conservative and fail to capture implicit or context-dependent expressions of subjectivity and unverifiability. Classical machine-learning models based on TF–IDF representations achieve moderate performance improvements, with F1-scores for the positive class ranging from 0.47 to 0.58. These models benefit from statistical learning but remain constrained by surface-level lexical features, limiting their ability to capture deeper semantic relationships in requirements text.

In contrast, the transformer-based DistilBERT model demonstrates strong performance, achieving weighted F1-scores of 0.74 for subjectivity and 0.86 for unverifiability. This indicates its effectiveness in capturing contextual and semantic information, particularly for identifying implicit linguistic patterns. However, transformer-based models do not explicitly incorporate normative linguistic rules defined in requirements engineering standards,

which may limit interpretability in safety-critical applications. The feature-level hybrid model further improves subjectivity detection, increasing the weighted F1-score from 0.74 to 0.79, while maintaining comparable performance for unverifiability (0.86). This improvement suggests that combining rule-based linguistic indicators with contextual embeddings enables the model to leverage both explicit domain knowledge and contextual understanding. The integration of symbolic and contextual features allows the hybrid model to better handle class imbalance and detect subtle subjective expressions.

Despite these findings, several limitations should be acknowledged. The dataset consists of 985 requirements collected from publicly available sources, which may introduce domain-specific bias and limit generalizability to industrial settings. The absence of formal inter-annotator agreement restricts the ability to quantify annotation consistency. Additionally, the use of a single train-test split limits the assessment of performance variability compared to cross-validation approaches. Transformer-based and hybrid models also require higher computational resources, introducing a trade-off between performance and efficiency.

From a practical perspective, the results suggest that transformer-based and hybrid approaches can significantly enhance automated requirement quality assessment, particularly in large-scale environments where manual inspection is costly. The inclusion of rule-based features in the hybrid model also supports interpretability during requirement validation. From a research perspective, the findings highlight the effectiveness of feature-level hybridization and provide a foundation for extending this approach to additional requirement smells such as ambiguity, inconsistency, and completeness. Future work may explore larger and more diverse datasets, incorporate cross-validation, and investigate explainability techniques to further improve robustness and interpretability.

Conclusion:

This study presented a controlled comparative evaluation of rule-based, classical machine-learning, transformer-based, and feature-level hybrid approaches for detecting subjectivity and unverifiability in natural language software requirements under identical experimental conditions. The results demonstrate consistent performance differences across detection paradigms, with transformer-based models outperforming rule-based and classical approaches in capturing contextual semantics. The proposed feature-level hybrid model improves subjectivity detection by combining rule-based linguistic indicators with contextual embeddings, while maintaining strong performance for unverifiability detection. These findings highlight the importance of integrating symbolic and contextual information to enhance detection performance, particularly under class imbalance conditions.

Overall, this study provides a unified experimental framework for evaluating requirement smell detection approaches and demonstrates the effectiveness of hybrid modeling strategies. Future work may extend this framework to additional requirement smells, evaluate performance on larger datasets, and incorporate explainability techniques to further improve interpretability and practical adoption.

References:

[1] "Natural Language Processing In Requirements Engineering And Its Challenges For Requirements Modelling In The Engineering Design Domain." Accessed: Apr. 21, 2026. [Online]. Available: https://www.researchgate.net/publication/371694672_NATURAL_LANGUAGE_PROCESSING_IN_REQUIREMENTS_ENGINEERING_AND_ITS_CHALLENGES_FOR_REQUIREMENTS_MODELLING_IN_THE_ENGINEERING_DESIGN_DOMAIN

[2] M. Tukur, S. Umar, and J. Hassine, "Requirement Engineering Challenges: A Systematic Mapping Study on the Academic and the Industrial Perspective," *Arab. J. Sci. Eng. 2021 464*, vol. 46, no. 4, pp. 3723–3748, Jan. 2021, doi: 10.1007/s13369-020-

05159-1.
[3] H. Femmer, D. Méndez Fernández, S. Wagner, S. Eder, "Rapid quality assurance with Requirements Smells," *arXiv:1611.08847*, 2016, [Online]. Available: https://arxiv.org/abs/1611.08847
[4] D. M. Fernández *et al.*, "Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice," *Empir. Softw. Eng.*, vol. 22, no. 5, pp. 2298–2338, Oct. 2017, doi: 10.1007/S10664-016-9451-7.
[5] H. Villamizar, T. Escovedo, and M. Kalinowski, "Requirements Engineering for Machine Learning: A Systematic Mapping Study," *Proc. - 2021 47th Euromicro Conf. Softw. Eng. Adv. Appl. SEAA 2021*, pp. 29–36, Sep. 2021, doi: 10.1109/SEAA53835.2021.00013.
[6] Alvaro Veizaga, Seung Yeob Shin, Lionel C. Briand, "Automated Smell Detection and Recommendation in Natural Language Requirements," *arXiv:2305.07097*, 2023, [Online]. Available: https://arxiv.org/abs/2305.07097
[7] V. Gervasi, A. Ferrari, D. Zowghi, and P. Spoletini, "Ambiguity in Requirements Engineering: Towards a Unifying Framework," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11865 LNCS, pp. 191–210, 2019, doi: 10.1007/978-3-030-30985-5_12.
[8] "ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering," Art. no. 29148–2011, Oct. 2018, doi: 10.1109/IEEESTD.2018.8559686.
[9] "(PDF) An Automatic Quality Evaluation for Natural Language Requirements." Accessed: Mar. 30, 2026. [Online]. Available: https://www.researchgate.net/publication/244206643_An_Automatic_Quality_Evaluation_for_Natural_Language_Requirements
[10] A. Ferrari *et al.*, "Detecting requirements defects with NLP patterns: an industrial experience in the railway domain," *Empir. Softw. Eng. 2018 236*, vol. 23, no. 6, pp. 3684–3733, Feb. 2018, doi: 10.1007/s10664-018-9596-7.
[11] M. Q. Riaz, W. H. Butt, and S. Rehman, "Automatic Detection of Ambiguous Software Requirements: An Insight," *5th Int. Conf. Inf. Manag. ICIM 2019*, pp. 1–6, May 2019, doi: 10.1109/INFOMAN.2019.8714682.
[12] V. Patel, P. Mehta, and K. Lavingia, "Software Requirement Classification Using Machine Learning Algorithms," *2023 Int. Conf. Artif. Intell. Appl. ICAIA 2023 Alliance Technol. Conf. ATCON-1 2023 - Proceeding*, 2023, doi: 10.1109/ICAIA57370.2023.10169588.
[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805*, 2018, [Online]. Available: https://arxiv.org/abs/1810.04805
[14] Israr Ali, Syed Sajjad Hussain Rizvi, "Enhancing Software Quality with AI: A Transformer-Based Approach for Code Smell Detection," *Appl. Sci.*, vol. 15, no. 8, p. 4559, 2025, doi: 10.3390/app15084559.
[15] A. Rahali and M. A. Akhloufi, "End-to-End Transformer-Based Models in Textual-Based NLP," *AI 2023, Vol. 4, Pages 54-110*, vol. 4, no. 1, pp. 54–110, Jan. 2023, doi: 10.3390/AI4010004.
[16] J. Peer, Y. Mordecai, and Y. Reich, "NLP4ReF: Requirements Classification and Forecasting: From Model-Based Design to Large Language Models," *IEEE Aerosp. Conf. Proc.*, 2024, doi: 10.1109/AERO58975.2024.10521022.
[17] Sallam Abualhaija, Chetan Arora, Mehrdad Sabetzadeh, Lionel C. Briand & Michael Traynor, "Automated demarcation of requirements in textual specifications: a machine learning-based approach," *Empir. Softw. Eng.*, vol. 25, pp. 5454–5497, 2020,

[Online]. Available: https://link.springer.com/article/10.1007/s10664-020-09864-1
[18] I. P. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, "Attention Is All You Need," *arXiv:1706.03762*, 2017, doi: https://doi.org/10.48550/arXiv.1706.03762.
[19] Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, Sivanesan Sangeetha, "AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing," *arXiv:2108.05542*, 2021, [Online]. Available: https://arxiv.org/abs/2108.05542
[20] Nadia Mushtaq Gardazi, Ali Daud, Muhammad Kamran Malik, Amal Bukhari, Tariq Alsahfi, "BERT applications in natural language processing: a review," *Artif. Intell. Rev.*, vol. 58, no. 166, 2025, [Online]. Available: https://link.springer.com/article/10.1007/s10462-025-11162-5
[21] Anna Rogers, Olga Kovaleva, Anna Rumshisky, "A Primer in BERTology: What we know about how BERT works," *arXiv:2002.12327*, 2020, [Online]. Available: https://arxiv.org/abs/2002.12327
[22] Ashagrew Liyih Alem, Ketema Keflie Gebretsadik, Shegaw Anagaw Mengistie & Muluye Fentie Admas, "Multi-label software requirement smells classification using deep learning," *Sci. Rep.*, 2025, [Online]. Available: https://www.nature.com/articles/s41598-025-86673-w
[23] M. K. Habib, S. Wagner, and D. Graziotin, "Detecting Requirements Smells with Deep Learning: Experiences, Challenges and Future Work," *Proc. IEEE Int. Conf. Requir. Eng.*, vol. 2021-September, pp. 153–156, Sep. 2021, doi: 10.1109/REW53955.2021.00027.
[24] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, Jianfeng Gao, "Deep Learning Based Text Classification: A Comprehensive Review," *arXiv:2004.03705*, 2021, [Online]. Available: https://arxiv.org/abs/2004.03705
[25] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," *Adv. Neural Inf. Process. Syst.*, vol. 2020-December, May 2020, Accessed: Sep. 26, 2024. [Online]. Available: https://arxiv.org/abs/2005.14165v4
[26] S. Ezzini, S. Abualhaija, C. Arora, M. Sabetzadeh, and L. C. Briand, "Using domain-specific corpora for improved handling of ambiguity in requirements," *Proc. - Int. Conf. Softw. Eng.*, pp. 1485–1497, Nov. 2021, doi: 10.1109/ICSE43902.2021.00133.
[27] Saad Ezzini, Sallam Abualhaija, Chetan Arora, Mehrdad Sabetzadeh, "TAPHSIR: Towards AnaPHoric Ambiguity Detection and ReSolution In Requirements," *arXiv:2206.10227*, 2022, [Online]. Available: https://arxiv.org/abs/2206.10227
[28] A. Fantechi, S. Gnesi, and L. Semini, "Rule-based NLP vs ChatGPT in ambiguity detection, a preliminary study," *REFSQ Work.*, 2023.
[29] Qixiang Zhou, Tong Li, "Assisting in requirements goal modeling: a hybrid approach based on machine learning and logical reasoning," *Proc. - 25th ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Model. 2022*, 2022, [Online]. Available: https://dl.acm.org/doi/10.1145/3550355.3552415
[30] "ERTMS/ETCS System Requirements Specification", [Online]. Available: https://www.era.europa.eu/system/files/2023-01/sos1_index001_-_era_ertms_003204_v500.pdf