BUILDING AND VALIDATING A SILVER-STANDARD BILINGUAL DATASET FOR MULTILABEL SOFTWARE REQUIREMENTS CLASSIFICATION

Truong Dac Dien and Nguyen Thi Xuan Huong

Faculty of Software Engineering University of Information Technology – VNUHCM Ho Chi Minh City, Vietnam

ABSTRACT

This paper presents a reproducible workflow for building and validating a silver-standard bilingual dataset for multilabel software requirements classification. The corpus contains 8,832 aligned English-Vietnamese requirement pairs with 12 labels, while a separate gold benchmark contains 622 aligned pairs. A blind audit of 500 silver instances gives a macro-averaged Cohen's kappa of 0.7427, indicating generally reliable annotations, with lower agreement for broader labels such as Look & Feel and Operability. Classical and transformer-based models are trained on the silver corpus, with threshold tuning performed only on silver validation folds. On the English gold benchmark, RoBERTa-base obtains the best micro-F1 without tuning (0.7766 ± 0.0040). On Vietnamese gold, PhoBERT-base-v2 performs best after tuning (0.7433 ± 0.0091). Threshold tuning benefits classical models more than transformers. Overall, silver data supports scalable training, but gold data remains necessary for reliable comparison.

KEYWORDS

Requirements engineering, multilabel classification, silver-standard dataset, bilingual dataset, threshold calibration.

1. INTRODUCTION

Requirements classification is an important task in requirements engineering, as it supports traceability, prioritization, compliance analysis, and downstream quality assurance [1, 2]. However, in practice, requirement labeling remains costly and time-consuming, as it typically relies on manual expert effort.

Advances in NLP have significantly benefited requirements engineering, yet two main challenges remain. First, most high-quality datasets are available in English, leaving low-resource languages such as Vietnamese with limited benchmarks [1, 2]. Second, research often emphasizes model architectures rather than annotation quality, threshold tuning, and the proper separation of training and test data [3, 4]. To address these issues, this study adopts a two-dataset approach built from PROMISE resources. For final testing, we use PROMISE-relabeled-NICE, a curated benchmark of 622 requirements derived from the PROMISE NFR collection [6, 7]. For silver construction, we use a larger pool of 8,832 unlabeled PROMISE requirements as the source material for automatic annotation. The resulting silver corpus is then used for model development in the silver regime, while the gold dataset is reserved for final external evaluation. Our main contributions are as follows:

DOI: 10.5121/ijnlc.2026.15201

- We construct and validate a bilingual silver corpus of 8,832 aligned English-Vietnamese requirement pairs, annotated with 12 labels.
- A bilingual gold benchmark of 622 aligned requirements is established and kept strictly separate from both training and threshold selection.
- A retrieval-augmented LLM annotation workflow is developed with schema-constrained outputs, bounded retries, and resume-safe execution to support reproducible long-running labeling processes [5], [11].
- A blind human audit of 500 silver instances is conducted, with label-wise Cohen's kappa scores reported to highlight reliable categories and those needing further refinement.
- Both classical and transformer approaches are evaluated using strict 10-fold cross-validation to avoid data leakage. The results show that threshold tuning substantially benefits classical methods, while its effect on transformer-based approaches remains limited.

The novelty of this work is not solely derived from the reuse of the PROMISE dataset. Its main contribution is the controlled integration of four components: a bilingual English–Vietnamese silver corpus, a separate gold benchmark, an auditable retrieval-augmented LLM labeling workflow, and a direct comparison between performance on silver-test and gold-test data. This design clearly defines the role of each dataset and reduces the common risk of using the same benchmark for both model development and final evaluation.

Compared with existing studies based on the PROMISE dataset, this work shifts the focus from developing a single classifier to designing a data construction and validation workflow. It examines whether silver-labeled data can support scalable training and whether a smaller gold benchmark is still necessary for reliable evaluation. This distinction is particularly important for low-resource languages, where fully manual annotation is often too costly.

The remainder of this paper is organized as follows. Section 2 reviews related work, Section 3 presents the dataset construction process, annotation workflow, and evaluation protocol, Section 4 reports the experimental results and discusses the respective roles of the silver and gold datasets, and Section 5 concludes the paper and outlines directions for future work.

2. RELATED WORK

Natural language has long been the main medium for writing software requirements, which explains why NLP has become a major research direction in requirements engineering. Zhao et al. provide a broad mapping of NLP for requirements engineering and show that requirements analysis, defect detection, and traceability remain central application areas [1]. Necula et al. also confirm the continued development of the field, while emphasizing the need for stronger empirical resources and more consistent evaluation practices [2].

Benchmark datasets for requirements classification remain limited. PROMISE-relabeled-NICE contains 622 curated requirements derived from the PROMISE NFR collection and is often treated as a gold-style benchmark because its labels were revised for consistency [6, 7]. Its relatively small size, however, makes it less suitable for training data-intensive models. This limitation motivates the use of silver-standard data when large manually curated corpora are not available.

Some recent studies show that traditional machine learning approaches remain effective for requirements classification. For example, Or applies SMOTE-Tomek preprocessing to the

PROMISE dataset and reports that logistic regression improves significantly when handling class imbalance [4]. This suggests that data handling and evaluation strategies can be as important as the choice of model.

Transformer-based approaches also provide strong baselines for this task. NoRBERT adapts BERT for requirements classification and achieves strong performance on PROMISE-style data [3]. Among the models used in this study, PhoBERT offers Vietnamese-specific pretraining [9], RoBERTa serves as a strong English baseline [8], and DeBERTaV3 improves representation learning through ELECTRA-style pretraining and disentangled embeddings [10].

A remaining gap in the literature is the lack of a clear separation between large-scale silver supervision and small-scale gold evaluation. This study addresses that gap by assigning distinct roles to each dataset and by examining how models trained on silver data transfer to evaluation on gold-only benchmarks.

Multi-label classification has a well-established methodological foundation. Grigorios Tsoumakas et al. outline key problem transformation and algorithm adaptation approaches, while Min-Ling Zhang and Zhi-Hua Zhou review major algorithms and evaluation challenges in multi-label learning. These works support our use of micro-F1, macro-F1, Hamming loss, and label-wise threshold tuning. In requirements engineering, public datasets such as PURE highlight the importance of reusable data and transparent curation. Compared with these studies, our work focuses on a bilingual silver–gold setting for requirements classification.

3. METHODOLOGY

3.1. Task Definition and Label Space

We formulate this task as a multilabel text classification problem, where each requirement is represented by a 12-dimensional binary vector. The label set includes Functional, Availability, Fault Tolerance, Legal, Look and Feel, Maintainability, Operability, Performance, Portability, Scalability, Security, and Usability. The label order is kept consistent across all experiments to ensure clarity and reproducibility. Following prior work such as Or [4], the labels are defined as follows:

- Functional (F): What the system needs to do and the services it provides.
- Availability (A): Rules to keep the system running and accessible.
- Fault Tolerance (FT): Rules to make sure the system works even if errors happen.
- Legal (L): Rules based on laws, contracts, or compliance.
- Look & Feel (LF): Requirements about the interface and how users see the system.
- Maintainability (MN): Requirements for fixing, updating, or changing the system over time.
- Operability (O): Rules for managing and running the system effectively.
- Performance (PE): Limits on speed, response time, or resource usage.
- Portability (PO): The ability to run the system on different platforms.
- Scalability (SC): How well the system handles more users or workload.
- Security (SE): Protecting the system and data from attacks.
- Usability (US): Making the system easy to learn and use.

Because this study treats requirements classification as a multilabel task, one requirement may receive more than one active label.

Table 1 gives representative examples from the project data and audit sample. It includes clear single-label requirements, multi-label requirements, and ambiguous boundary cases.

Table 1. Representative clear, multi-label, and ambiguous requirements in the multilabel setting.

Requirement example	**Active labels**	**Annotation note**
The system shall provide a history report of changes made to the Activity or Event data.	Functional (F)	Clear single-label case: the requirement describes a concrete system service.
Only registered realtors shall be able to access the system.	Security (SE)	Clear single-label case: the main intent is access control.
The product is expected to run on Windows CE and Palm operating systems.	Portability (PO)	Clear single-label case: the requirement specifies supported operating systems.
The system shall refresh the display every 60 seconds.	Functional (F); Performance (PE)	Multi-label case: the requirement asks for a system action and gives a timing constraint.
The application shall match the color of the schema set forth by Department of Homeland Security.	Legal (L); Look & Feel (LF)	Multi-label case: the color scheme is both a compliance constraint and a visual-design constraint.
The product shall ensure that it can only be accessed by authorized users. The product will be able to distinguish between authorized and unauthorized users in all access attempts.	Functional (F); Security (SE)	Multi-label case: the statement describes an access-control function and a security constraint.
The product shall be capable of handling the existing 1000 users. This number is expected to grow 5 times within the next year.	Performance (PE); Scalability (SC)	Multi-label case: the requirement concerns current load handling and future growth.
Policies and procedures for communicating information to appropriate stakeholders should be clearly defined.	Legal (L)	Ambiguous audit case: policy wording may be confused with Operability, but the human audit treats it as a compliance obligation.
The platform should support multilingual interfaces.	Functional (F); Usability (US)	Ambiguous audit case: language support was confused with Portability, but the human audit treats it as a feature that improves user access.

The added examples make the scope of Table 1 explicit. Clear cases show requirements with one dominant label, multi-label cases show requirements with more than one active label, and ambiguous audit cases show requirements where the boundary between labels is difficult.

The examples in Table 1 illustrate the need for clear label boundaries. Timing expressions typically indicate Performance, interface-related wording may reflect Usability or Look & Feel depending on the intended meaning, and policy-related statements can be confused with Operability unless their compliance role is explicitly defined.

These examples also highlight why multi-label requirements classification is more challenging than traditional single-label text classification. A single requirement may describe a system function while simultaneously imposing a quality constraint. Therefore, we allow multiple active labels and report both micro-level and macro-level evaluation metrics.

3.2. Dataset Construction and Harmonization

We use two datasets in this study:

- Silver modeling corpus: This dataset contains 8,832 aligned English-Vietnamese requirement pairs and is used for training, validation, threshold tuning, and internal evaluation.
- Gold benchmark: This dataset includes 622 English-Vietnamese pairs derived from PROMISE-relabeled-NICE and is reserved exclusively for final evaluation.

Both datasets share the same structure, consisting of requirement text and corresponding labels. Before training, we checked the data for format consistency and a fixed label order. The silver corpus is aligned at the sentence level across English and Vietnamese. No exact duplicates were found in the English data, while nine duplicates were found in the Vietnamese version. These duplicates were retained to preserve strict alignment between the two languages. Table 2 summarizes the composition of the corpora and the role of each resource.

Table 2. Corpus composition and experimental role of each resource.

Dataset	Language	Samples	Labels	Split usage
Silver	EN	8832	12	train/val/test (silver)
Silver	VI	8832	12	train/val/test (silver)
Gold	EN	622	12	test gold
Gold	VI	622	12	test gold

This distinction is important throughout the paper. When referring to a model trained on silver, we mean training on the 8,832-instance silver corpus. When referring to testing on gold, we mean evaluation on the 622-instance gold benchmark, which is not used for training or threshold tuning.

Figure1 illustrates the overall pipeline, from raw text collection to the release of the final bilingual dataset.

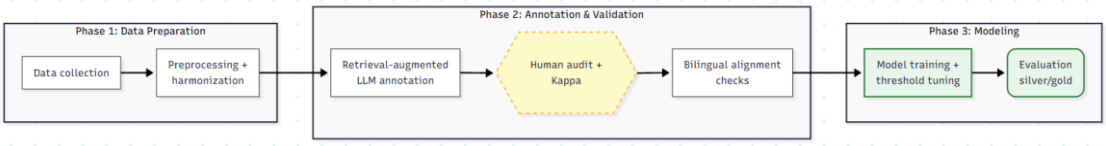

Figure 1. End-to-end workflow from raw requirements to the final bilingual silver modeling corpus and gold benchmark.

3.3. Label Distribution and Imbalance Analysis

The most frequent label is Functional, with 6,355 positive instances (71.95%), while the least frequent label is Scalability, with 221 positives (2.50%). The imbalance ratio is defined as follows:

$$\mathrm{IR} = \frac{max_\ell N_\ell}{min_\ell N_\ell} = \frac{6355}{221} = 28.76$$

Here, N_ℓ denotes the number of positive instances for label ℓ. Among the non-functional categories, Security (1,400) and Operability (1,045) have the highest frequencies. This imbalance

highlights the need for macro-F1 and label-wise agreement, as micro-F1 alone would be dominated by the Functional label.
Because the bilingual resources are aligned at the instance level, English and Vietnamese share identical label distributions within each dataset. As a result, a single distribution plot for the silver corpus and one for the gold benchmark are sufficient.

Figures2 and 3 present the label distributions for the final silver modeling corpus and the gold benchmark, respectively.

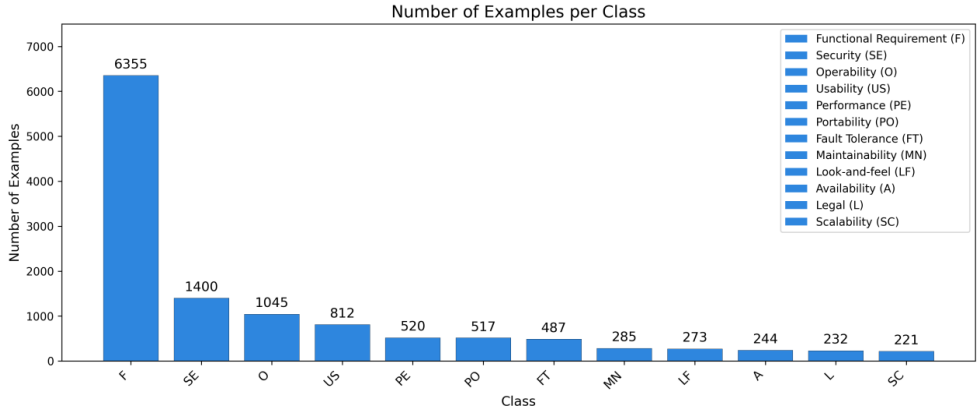

Figure 2. Label frequency distribution in the final silver modeling corpus.

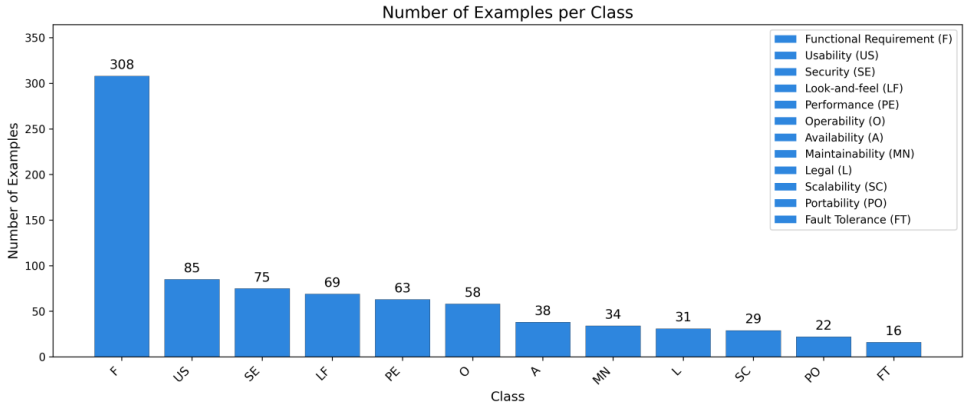

Figure 3. Label frequency distribution in the bilingual gold benchmark.

Figures 2 and 3 illustrate the presence of label imbalance. Functional requirements dominate both corpora, while labels such as Scalability and Look & Feel appear infrequently. Although this distribution reflects realistic software requirements, it creates a risk that models perform better on frequent labels than on rare ones. Therefore, the evaluation includes macro-F1 and per-label kappa in addition to micro-F1.

3.4. Retrieval-Augmented LLM Annotation

We generate silver labels using a retrieval-augmented LLM workflow designed to ensure reproducibility. For each unlabeled requirement selected for annotation, the system retrieves

similar reference examples using TF-IDF and cosine similarity, and then prompts the LLM to produce a schema-constrained multilabel decision [5, 11].
Each output row includes three components: (i) a brief rationale, (ii) 12 binary label assignments, and (iii) a confidence score between 0 and 1. This structured format supports systematic validation at the row level and reduces ambiguity in later stages.

To handle long-running processes, we use controlled concurrency, retries with exponential backoff, and regular checkpointing, allowing the workflow to resume from specific IDs. The prompts are designed to ensure that the model assigns labels consistently and adheres to the required format. If an invalid output is produced, the system automatically retries under the same constraints.

We also record model provenance, retrieval settings, confidence scores, and rationales to ensure that the annotation process can be audited or partially rerun when needed.

The prompt consists of three parts. First, it presents the definitions of the 12 labels, along with typical keywords and decision boundaries, such as Availability versus Fault Tolerance and Usability versus Look & Feel. Second, it includes up to four nearest PROMISE examples retrieved using TF-IDF with 1–2 grams and cosine similarity. Third, it instructs the model to return a fixed schema containing a brief rationale, 12 Boolean labels, and a confidence score between 0 and 1.

The annotation run used gpt-5-mini with low reasoning effort, periodic checkpointing every 25 completed rows, and eight parallel requests. The output parser rejected invalid structures, and temporary API errors were handled with up to five attempts using exponential backoff. These controls do not make the labels perfect, but they make the labeling process traceable, repeatable, and easier to audit.

A compact form of the user prompt is as follows: “Given the target requirement and the retrieved reference examples, analyze the intent of the requirement and assign each of the 12 labels as true or false. Use true only when there is explicit evidence.” This conservative instruction is important because over-labeling would increase recall while reducing precision.

3.5. Human Reliability Validation

To assess label reliability, a human reviewer evaluated a random sample of 500 requirements from the silver dataset. The reviewer was provided only with the ID and the text, without access to the generated labels.

Agreement is measured with label-wise Cohen's kappa [13]:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

Where p_o is the observed agreement and p_e is the agreement expected by chance.

The overall macro-kappa score is 0.7427. Four labels scored higher than 0.80, six were between 0.67 and 0.80, and two were under 0.60. This shows that the silver labels are generally reliable, although quality varies across categories. For example, Look & Feel and Operability are more difficult to label consistently.

Table3 reports the label-wise agreement scores together with human and LLM positive counts.

Table 3. Per-label Cohen's kappa on the 500-instance blind audit.

Label	Positive Human	Positive LLM	Cohen's Kappa
Functional (F)	408	368	0.738
Availability (A)	12	12	1.000
Fault Tolerance (FT)	20	30	0.706
Legal (L)	15	19	0.757
Look & Feel (LF)	5	13	0.549
Maintainability (MN)	16	20	0.770
Operability (O)	31	74	0.405
Performance (PE)	19	21	0.896
Portability (PO)	23	29	0.676
Scalability (SC)	4	7	0.724
Security (SE)	69	68	0.873
Usability (US)	56	49	0.819

This audit is best interpreted as human-AI agreement, not as multi-rater human inter-annotator agreement. Therefore, the results support the practical usability of the silver corpus, but they do not replace a future multi-annotator study.

For a total of 8,832 samples, standard statistical formulas (95% confidence level and 5% margin of error) indicate that at least 368 samples should be reviewed. By evaluating 500 samples, we comfortably exceed this requirement.

The two weakest labels are Operability ($\kappa = 0.405$) and Look & Feel ($\kappa = 0.549$).Operability is broad, as it includes configuration, monitoring, administration, and deployment management. Look & Feel is rare and often overlaps with Usability when interface-related wording does not clearly distinguish visual design from ease of use. These categories are therefore treated as priority targets for future annotation refinement.

Because the audit involved a single human reviewer, the kappa values should not be interpreted as full inter-annotator agreement. Instead, they measure the agreement between the LLM-generated labels and a blind human audit. This is still useful for assessing the quality of silver labels; however, a more robust gold-standard validation would require at least two independent human annotators, conflict adjudication, and a formal annotation guideline.

3.6. Bilingual Corpus Alignment

We ensure that both datasets are fully aligned between English and Vietnamese. The number of instances and label indices are identical across the two languages. Although the wording differs, the underlying meaning is preserved. In the silver dataset, English requirements average 111.31 characters and 17.52 words, while Vietnamese requirements average 109.53 characters and 24.22 words. This difference is expected, as Vietnamese uses more spaces between syllables.

Such alignment is essential for fair cross-lingual comparison and enables traceable error analysis between corresponding English and Vietnamese requirements.

3.7. Experimental Design

We evaluate the models under two settings:

- Silver regime: Models are trained and evaluated within the 8,832-instance silver corpus.
- Gold regime: Models trained on silver data are evaluated on the 622-instance gold benchmark.

The two datasets are kept strictly separate. The silver corpus is used for training, threshold tuning, and intermediate evaluation, while the gold benchmark is reserved exclusively for final comparison.

We compare five traditional approaches (Naive Bayes, Logistic Regression, Linear SVM, Random Forest, and CatBoost) with transformer-based models (RoBERTa-base and DeBERTaV3-base for English; PhoBERT-base and PhoBERT-base-v2 for Vietnamese).
For the silver corpus, we apply 10-fold cross-validation [14]. In each fold, the data is split into training, validation, and test subsets. Thresholds for each label are tuned using only the validation set by searching values from 0.05 to 0.95 to maximize the F1 score. Each fold follows these steps:

- Train the model on the silver training partition.
- Tune per-label thresholds on the silver validation partition only.
- Report in-domain performance on the silver test partition.
- Evaluate the same trained model once on the full gold benchmark to measure transfer from silver supervision to gold evaluation.

This design makes the role of each dataset explicit. The silver test split addresses the question: "How well does the model perform on data from the same silver distribution?" In contrast, the gold benchmark evaluates a different aspect: "How well does a model trained on silver data transfer to a smaller, curated benchmark?"

Classical baselines are trained using a one-vs-rest strategy with TF-IDF features (1-2 grams, vocabulary capped at 50,000, minimum document frequency 1). Transformer models are configured with a maximum sequence length of 256, batch size of 8, 5 training epochs, learning rate 2e-5, weight decay 0.01, warmup ratio of 0.1, default threshold of 0.5, and a fixed random seed of 42. Both model groups use identical fold assignments to ensure a fair comparison.

After final filtering, the risk of contamination is minimal. An exact-text overlap audit between the silver corpus and the gold benchmark shows 0/622 overlaps for English and 3/622 for Vietnamese. We report this explicitly, as even small overlaps should be documented in requirements NLP experiments.

3.8. Evaluation Metrics

We report four standard multilabel metrics: micro-F1, macro-F1, Hamming loss, and subset accuracy.

To avoid notation ambiguity, we define all symbols in the text. Let n be the number of instances and L be the number of labels. For instance$i$ and label ℓ, $y_{i\ell}$denotes the ground-truth valueand

$\hat{y}_{i\ell}$denotes the predicted value. For each label ℓ, TP_ℓ, FP_ℓ, FN_ℓ, and TN_ℓrepresenttrue positives, false positives, false negatives, and true negatives, respectively, in a one-vs-rest setting.

$$P_{\text{micro}} = \frac{\sum_{\ell=1}^{L} T\,P_\ell}{\sum_{\ell=1}^{L}(T\,P_\ell + FP_\ell)}$$

$$R_{\text{micro}} = \frac{\sum_{\ell=1}^{L} T\,P_\ell}{\sum_{\ell=1}^{L}(T\,P_\ell + FN_\ell)}$$

$$F1_{\text{micro}} = \frac{2P_{\text{micro}}R_{\text{micro}}}{P_{\text{micro}} + R_{\text{micro}}}$$

$$F1_{\text{macro}} = \frac{1}{L}\sum_{\ell=1}^{L} F\,1_\ell$$

Hamming loss and subset accuracy are defined below. Hamming loss measures element-wise label errors, while subset accuracy requires the whole predicted label set to match the ground truth exactly.

$$\text{HammingLoss} = \frac{1}{nL}\sum_{i=1}^{n}\sum_{\ell=1}^{L}\mathbf{1}\left[y_{i\ell} \neq \hat{y}_{i\ell}\right]$$

$$\text{SubsetAccuracy} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left[y_i = \hat{y}_i\right]$$

Together, these metrics capture overall label quality (micro-F1), balance across labels (macro-F1), element-wise error rate (Hamming loss), and exact-match performance (subset accuracy).

4. RESULTS AND DISCUSSION

4.1. Reliability and Dataset Readiness

The review indicates that the silver dataset is generally useful, although quality varies across categories. A macro kappa of 0.7427 reflects a solid level of agreement. Labels such as Availability, Performance, Security, and Usability show high reliability, while Look & Feel (0.549) and Operability (0.405) perform less consistently. These latter categories are broader and often overlap with functional aspects, which makes them more difficult to classify.

Within our experimental design, the silver corpus supports model development in the silver regime, including training, validation, threshold tuning, and in-domain testing, while the gold benchmark is reserved for final external testing. At the same time, the weaker labels should be interpreted with caution, and improving their definitions and annotations remains important for raising overall model performance.

4.2. Silver Benchmark Results

Tables 4–7 report in-domain performance on the silver test folds. These tables address a different question from the gold benchmark: they show how well each model fits the large silver distribution after being trained and validated on other silver folds. The reported values are the mean and standard deviation across 10 folds.

In the English no-tuning setting, RoBERTa-base gives the strongest silver result with micro-F1 of 0.8511. DeBERTaV3-base reaches 0.8076, and Linear SVM reaches 0.8059. Thus, the transformer model is strongest, but the best sparse baseline is still competitive. The low macro-F1 of Naive Bayes shows that simple probabilistic models struggle with rare labels.
After threshold tuning, the English classical models show greater improvement than the transformer models. Linear SVM increases from 0.8059 to 0.8142 in micro-F1, while RoBERTa-base changes from 0.8511 to 0.8466. This suggests that threshold tuning mainly benefits models for which the default 0.5 cutoff is not well suited to imbalanced labels.

For Vietnamese, PhoBERT-base-v2 is the strongest model on the silver test folds, reaching 0.8266 without tuning and 0.8257 with tuning. Linear SVM is the strongest classical baseline. The small gap between PhoBERT-base-v2 and Linear SVM suggests that a well-tuned sparse model can still be competitive when the requirement wording is regular and the label set is stable.

The key interpretation is that results on the silver test set measure internal consistency rather than true external validity. High performance on silver data indicates that the silver labels are useful for scalable training. However, final conclusions still require the gold benchmark, as it is manually curated and kept separate from both training and threshold tuning.

Table 4. English silver evaluation without label-wise threshold tuning. The highest micro-F1 is shown in bold.

Model	**micro-F1**	**macro-F1**	**Hamming loss**	**Subset accuracy**
CatBoost	0.7481 ± 0.0062	0.4975 ± 0.0172	0.0522 ± 0.0015	0.5721 ± 0.0092
Linear SVM	0.8059 ± 0.0062	0.6205 ± 0.0168	0.0410 ± 0.0013	0.6366 ± 0.0155
Logistic Regression	0.6937 ± 0.0078	0.2255 ± 0.0215	0.0589 ± 0.0014	0.5220 ± 0.0135
Naive Bayes	0.6328 ± 0.0011	0.0910 ± 0.0061	0.0706 ± 0.0005	0.4705 ± 0.0082
Random Forest	0.7553 ± 0.0072	0.4642 ± 0.0185	0.0503 ± 0.0016	0.5858 ± 0.0106
DeBERTaV3-base	0.8076 ± 0.0091	0.6559 ± 0.0295	0.0435 ± 0.0021	0.6352 ± 0.0139
RoBERTa-base	**0.8511 ± 0.0075**	0.7469 ± 0.0198	0.0338 ± 0.0016	0.6942 ± 0.0110

Table 5. English silver evaluation with label-wise threshold tuning. The highest micro-F1 is shown in bold.

Model	**micro-F1**	**macro-F1**	**Hamming loss**	**Subset accuracy**
CatBoost	0.7669 ± 0.0124	0.6267 ± 0.0234	0.0539 ± 0.0035	0.5517 ± 0.0227
Linear SVM	0.8142 ± 0.0082	0.6951 ± 0.0189	0.0437 ± 0.0023	0.6212 ± 0.0142
Logistic Regression	0.7841 ± 0.0081	0.6323 ± 0.0222	0.0503 ± 0.0024	0.5817 ± 0.0163
Naive Bayes	0.7164 ± 0.0067	0.2678 ± 0.0145	0.0558 ± 0.0015	0.5361 ± 0.0133
Random Forest	0.7946 ± 0.0064	0.6683 ± 0.0107	0.0482 ± 0.0024	0.5929 ± 0.0155
DeBERTaV3-base	0.7991 ± 0.0130	0.6505 ± 0.0319	0.0472 ± 0.0033	0.6117 ± 0.0205
RoBERTa-base	**0.8466 ± 0.0071**	0.7457 ± 0.0182	0.0361 ± 0.0017	0.6774 ± 0.0094

Table 6. Vietnamese silver evaluation without label-wise threshold tuning. The highest micro-F1 is shown in bold.

Model	**micro-F1**	**macro-F1**	**Hamming loss**	**Subset accuracy**
CatBoost	0.7611 ± 0.0078	0.5369 ± 0.0242	0.0507 ± 0.0015	0.5814 ± 0.0102
Linear SVM	0.8103 ± 0.0103	0.6493 ± 0.0178	0.0411 ± 0.0021	0.6391 ± 0.0183

Logistic Regression	0.7530 ± 0.0093	0.4806 ± 0.0251	0.0512 ± 0.0020	0.5823 ± 0.0118
Naive Bayes	0.6432 ± 0.0046	0.1102 ± 0.0052	0.0691 ± 0.0013	0.4805 ± 0.0106
Random Forest	0.7588 ± 0.0100	0.4705 ± 0.0219	0.0500 ± 0.0019	0.5873 ± 0.0134
PhoBERT-base-v2	**0.8266 ± 0.0085**	0.6605 ± 0.0220	0.0384 ± 0.0019	0.6635 ± 0.0154
PhoBERT-base	0.8062 ± 0.0081	0.5479 ± 0.0409	0.0416 ± 0.0018	0.6327 ± 0.0134

Table 7. Vietnamese silver evaluation with label-wise threshold tuning. The highest micro-F1 is shown in bold.

Model	**micro-F1**	**macro-F1**	**Hamming loss**	**Subset accuracy**
CatBoost	0.7747 ± 0.0096	0.6345 ± 0.0246	0.0533 ± 0.0026	0.5575 ± 0.0204
Linear SVM	0.8117 ± 0.0094	0.6983 ± 0.0205	0.0445 ± 0.0025	0.6140 ± 0.0143
Logistic Regression	0.7660 ± 0.0085	0.6065 ± 0.0180	0.0563 ± 0.0029	0.5383 ± 0.0131
Naive Bayes	0.7077 ± 0.0084	0.2569 ± 0.0145	0.0576 ± 0.0015	0.5338 ± 0.0116
Random Forest	0.7990 ± 0.0084	0.6733 ± 0.0239	0.0472 ± 0.0022	0.5990 ± 0.0164
PhoBERT-base-v2	**0.8257 ± 0.0090**	0.6992 ± 0.0211	0.0408 ± 0.0026	0.6480 ± 0.0234
PhoBERT-base	0.8092 ± 0.0096	0.6564 ± 0.0301	0.0450 ± 0.0025	0.6198 ± 0.0200

4.3. Gold Benchmark Results

Tables 8-11 present the gold benchmark results for both languages. These results are more conservative than the silver-test results because the gold benchmark is smaller, manually curated, and never used for training or threshold selection. For English, RoBERTa-base achieves the highest gold micro-F1 score, reaching 0.7766 before threshold tuning and 0.7693 after tuning. For Vietnamese, PhoBERT-base-v2 performs best, with 0.7350 before tuning and 0.7433 after tuning.

Among the traditional approaches, Linear SVM delivers the strongest gold performance in both languages. Its Vietnamese micro-F1 of 0.7236 after tuning is close to the transformer results. This finding is important for practical use: transformer models lead overall, but Linear SVM remains a strong and efficient alternative when computing resources are limited.

The gold tables should be interpreted alongside the silver tables. The ranking of models is similar, but the scores are lower on the gold benchmark. This indicates that the silver corpus is useful for training, but does not eliminate the need for an independent benchmark. This is the main empirical reason for keeping the roles of the silver and gold datasets separate.

Table 8. English gold evaluation without label-wise threshold tuning. The highest micro-F1 is shown in bold.

Model	**micro-F1**	**macro-F1**	**Hamming loss**	**Subset accuracy**
CatBoost	0.5888 ± 0.0062	0.4089 ± 0.0132	0.0823 ± 0.0008	0.4445 ± 0.0034
Linear SVM	0.7197 ± 0.0030	0.5725 ± 0.0107	0.0544 ± 0.0003	0.5838 ± 0.0065
Logistic Regression	0.5250 ± 0.0048	0.1776 ± 0.0087	0.0844 ± 0.0009	0.4093 ± 0.0055
Naive Bayes	0.4479 ± 0.0013	0.0614 ± 0.0011	0.1019 ± 0.0004	0.3675 ± 0.0008
Random Forest	0.6212 ± 0.0045	0.4286 ± 0.0096	0.0745 ± 0.0007	0.4932 ± 0.0057
DeBERTaV3-	0.7122 ± 0.0101	0.6044 ± 0.0169	0.0628 ± 0.0021	0.5389 ± 0.0112

base				
RoBERTa-base	**0.7766 ± 0.0040**	0.6999 ± 0.0082	0.0483 ± 0.0009	0.6166 ± 0.0060

Table 9. English gold evaluation with label-wise threshold tuning. The highest micro-F1 is shown in bold.

Model	**micro-F1**	**macro-F1**	**Hamming loss**	**Subset accuracy**
CatBoost	0.6617 ± 0.0131	0.5815 ± 0.0164	0.0757 ± 0.0041	0.4453 ± 0.0209
Linear SVM	0.7514 ± 0.0060	0.6593 ± 0.0155	0.0543 ± 0.0018	0.5924 ± 0.0104
Logistic Regression	0.7112 ± 0.0105	0.6070 ± 0.0161	0.0619 ± 0.0033	0.5318 ± 0.0206
Naive Bayes	0.5534 ± 0.0059	0.2061 ± 0.0092	0.0794 ± 0.0009	0.4307 ± 0.0041
Random Forest	0.7207 ± 0.0074	0.6189 ± 0.0146	0.0609 ± 0.0016	0.5582 ± 0.0107
DeBERTaV3-base	0.7051 ± 0.0115	0.6045 ± 0.0148	0.0675 ± 0.0040	0.5121 ± 0.0231
RoBERTa-base	**0.7693 ± 0.0066**	0.6949 ± 0.0108	0.0522 ± 0.0020	0.5918 ± 0.0126

Table 10. Vietnamese gold evaluation without label-wise threshold tuning. The highest micro-F1 is shown in bold.

Model	**micro-F1**	**macro-F1**	**Hamming loss**	**Subset accuracy**
CatBoost	0.6061 ± 0.0062	0.4427 ± 0.0109	0.0796 ± 0.0011	0.4600 ± 0.0048
Linear SVM	0.7002 ± 0.0038	0.5923 ± 0.0069	0.0601 ± 0.0009	0.5587 ± 0.0072
Logistic Regression	0.5791 ± 0.0033	0.3559 ± 0.0063	0.0806 ± 0.0007	0.4535 ± 0.0037
Naive Bayes	0.4610 ± 0.0021	0.0890 ± 0.0025	0.1004 ± 0.0005	0.3717 ± 0.0007
Random Forest	0.6059 ± 0.0064	0.3967 ± 0.0143	0.0770 ± 0.0013	0.4781 ± 0.0050
PhoBERT-base-v2	**0.7350 ± 0.0073**	0.6335 ± 0.0100	0.0560 ± 0.0016	0.5696 ± 0.0084
PhoBERT-base	0.6801 ± 0.0093	0.4981 ± 0.0289	0.0647 ± 0.0016	0.5058 ± 0.0122

Table 11. Vietnamese gold evaluation with label-wise threshold tuning. The highest micro-F1 is shown in bold.

Model	**micro-F1**	**macro-F1**	**Hamming loss**	**Subset accuracy**
CatBoost	0.6535 ± 0.0058	0.5765 ± 0.0099	0.0789 ± 0.0029	0.4162 ± 0.0186
Linear SVM	0.7236 ± 0.0082	0.6622 ± 0.0135	0.0618 ± 0.0030	0.5294 ± 0.0160
Logistic Regression	0.6425 ± 0.0057	0.5437 ± 0.0090	0.0804 ± 0.0037	0.4214 ± 0.0250
Naive Bayes	0.5432 ± 0.0063	0.2444 ± 0.0141	0.0831 ± 0.0010	0.4122 ± 0.0042
Random Forest	0.7038 ± 0.0077	0.6289 ± 0.0168	0.0661 ± 0.0020	0.5050 ± 0.0135
PhoBERT-base-v2	**0.7433 ± 0.0091**	0.6703 ± 0.0132	0.0576 ± 0.0029	0.5619 ± 0.0165
PhoBERT-base	0.7219 ± 0.0107	0.6435 ± 0.0164	0.0636 ± 0.0034	0.5079 ± 0.0230

4.4. Effect of Threshold Calibration

Threshold calibration is analyzed after presenting the silver and gold benchmark results because the thresholds are selected on the silver validation folds. Therefore, the silver test results are discussed first, as they reflect the in-domain effect of tuning. The gold test results are then examined to assess whether the same tuning behavior generalizes to the independent benchmark.

Figures 4 and 5 show the change in silver-test micro-F1 after label-wise threshold tuning. In English, the largest improvements are observed for Logistic Regression (+0.0904), Naive Bayes (+0.0836), and Random Forest (+0.0392). CatBoost and Linear SVM show smaller gains, while

RoBERTa-base and DeBERTaV3-base decrease slightly. This pattern suggests that threshold tuning is most beneficial for classical models whose default decision cutoff is not well suited to imbalanced labels.

The Vietnamese silver-test results show a similar pattern. Naive Bayes (+0.0645) and Random Forest (+0.0402) achieve the largest improvements. CatBoost, Logistic Regression, Linear SVM, and PhoBERT-base change only slightly, while PhoBERT-base-v2 remains almost unchanged. These results indicate that threshold tuning can improve simple or sparse classifiers, but does not consistently benefit transformer-based models.

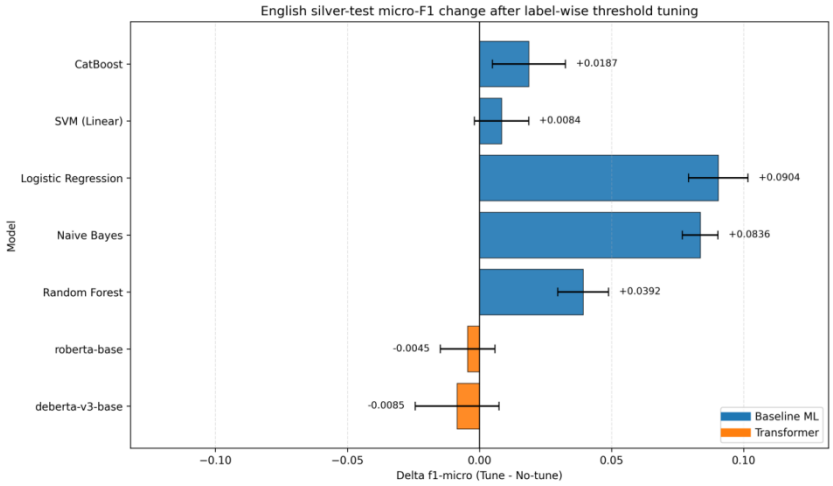

Figure 4. English silver-test micro-F1 change after label-wise threshold tuning.

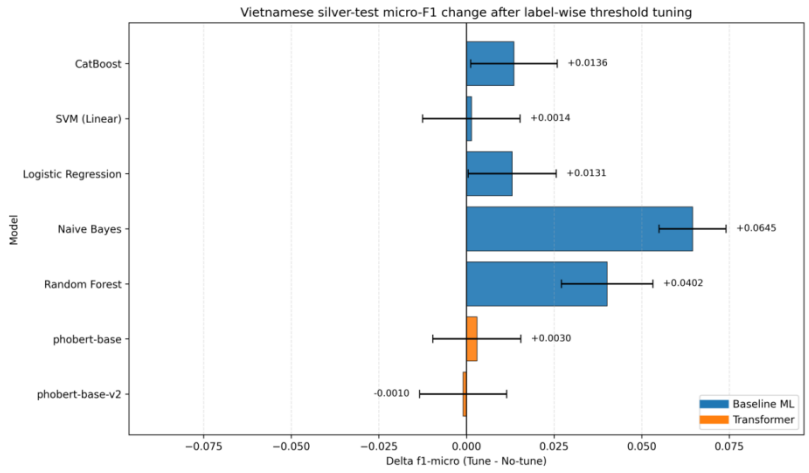

Figure 5. Vietnamese silver-test micro-F1 change after label-wise threshold tuning.

Figures 6 and 7 present the same analysis on the gold benchmark. In English, all classical models improve after threshold tuning, including Logistic Regression, Naive Bayes, Random Forest, CatBoost, and Linear SVM. In contrast, RoBERTa-base and DeBERTaV3-baseshow slight decreases. This result suggests that their score rankings already generalize well to the gold benchmark without additional calibration.

In the Vietnamese gold evaluation, threshold tuning also improves most classical baselines. Random Forest, Naive Bayes, Logistic Regression, CatBoost, and Linear SVM all show increases. PhoBERT-base also improves, while PhoBERT-base-v2 changes only slightly and remains the strongest Vietnamese model overall. This finding indicates that threshold tuning can improve the balance between recall and precision, but its effect depends on the model family.

Overall, the silver-first and gold-second analysis leads to a cautious conclusion. Threshold tuning is mainly beneficial for classical baselines under label imbalance, while its effects on transformer-based models are smaller and less consistent. Therefore, threshold tuning should be reported as a calibration step rather than a method that consistently improves all models.

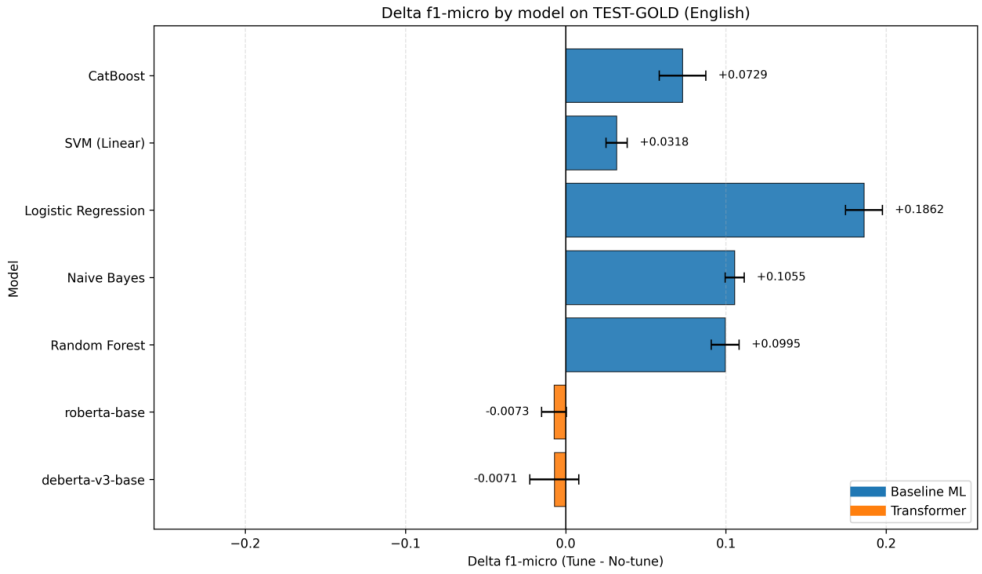

Figure 6. English gold-test micro-F1 change after label-wise threshold tuning.

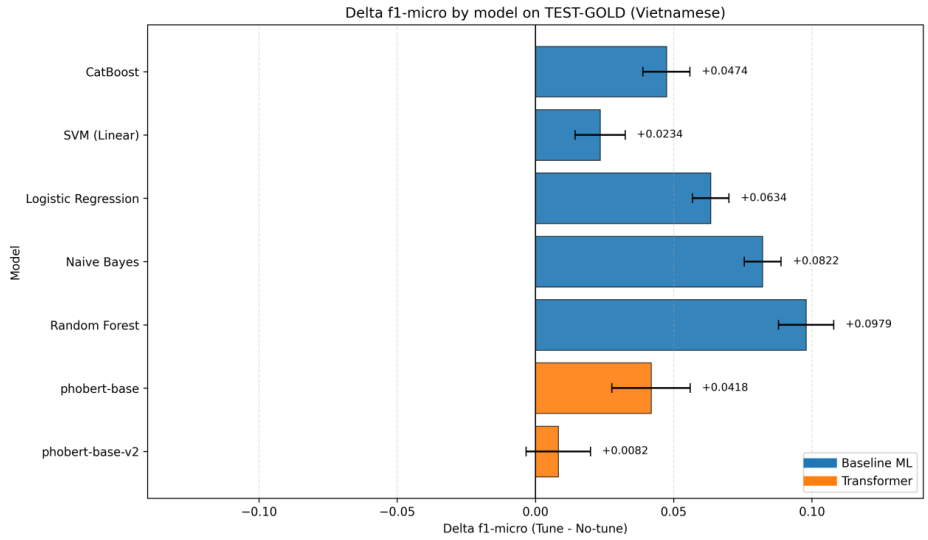

Figure 7. Vietnamese gold-test micro-F1 change after label-wise threshold tuning.

4.5. Silver-to-Gold Transferability

The comparison between silver and gold evaluation highlights the need for both datasets. The silver corpus is large enough to support model development and stable fold-based comparison, but performance on silver is generally higher than on external gold data. For example, the best English micro-F1 reaches 0.8511 on the silver test set, compared with 0.7766 on the gold benchmark. A similar pattern is observed in Vietnamese, where the best silver micro-F1 is 0.8266, while the best gold micro-F1 reaches 0.7433 after threshold tuning.

This gap should not be interpreted as a failure of the silver dataset. Instead, it reflects the different roles of the silver corpus and the gold benchmark. The silver corpus is suitable for scalable training, threshold selection, and early model comparison, while the gold benchmark remains necessary for final evaluation because it is cleaner, smaller, and closer to a controlled, human-validated reference set.

4.6. Cross-Language and Model-Family Observations

We observe that the best-performing transformer varies by language. RoBERTa-base performs best for English, while PhoBERT-base-v2 achieves the highest results for Vietnamese. This highlights the continued importance of language-specific pretraining, even for technical domains such as software requirements.

At the same time, classical approaches remain relevant. Linear SVM is the strongest baseline across all settings and is particularly competitive on the Vietnamese gold benchmark. These findings suggest that the conclusion is not "transformers only" or "classical models only." Instead, while transformers lead overall, a well-tuned sparse baseline can remain close enough to be a practical choice when simplicity, efficiency, or interpretability is important.

The English and Vietnamese results also show that translation does not make the task identical across languages. Vietnamese requirements often use more tokens to express the same meaning, as spaces may separate syllables. This can affect sparse features and tokenization. Transformer models help reduce this issue, but the difference between RoBERTa-base and PhoBERT-base-v2 indicates that language-specific pretraining remains important for technical requirements.

The comparison across model families offers a practical guideline. If the objective is maximum accuracy, transformer models are the preferred choice. If a transparent and cost-efficient baseline is required, Linear SVM remains a strong option. This is particularly relevant for small research teams or industrial settings with limited GPU resources.

4.7. Qualitative Error Analysis

The blind audit helps identify the main sources of error. The strongest categories are Availability, Performance, Security, and Usability, where the wording usually contains clear signals such as uptime, response time, access control, or ease of use. The weaker categories are Operability and Look & Feel.Operability is challenging because terms such as manage, configure, policy, and monitor may refer either to administrative tasks or to business rules. Look & Feel is also challenging, as interface-related wording can overlap with Usability.

Table 12. Representative disagreement cases from the 500-instance audit sample.

Requirement example	Human labels	LLM labels	Main issue
Policies and procedures for communicating information to appropriate stakeholders should be clearly defined.	Legal (L)	Operability (O)	Policy and compliance wording was confused with operational management.
The platform should support multilingual interfaces.	Functional (F); Usability (US)	Portability (PO)	The model interpreted language support as environment portability rather than user-facing access.
System interface must be compatible with specialist software and suitable for users with special needs.	Portability (PO); Usability (US)	Operability (O); Usability (US)	Compatibility was partly confused with operational integration.
User shall be able to navigate through the CCTNS application.	Functional (F); Usability (US)	Functional (F)	The usability aspect was implicit and therefore missed by the model.

The disagreement examples support this interpretation. The requirement "Policies and procedures for communicating information to appropriate stakeholders should be clearly defined" was marked as Operability by the LLM, but the human reviewer marked it as Legal because it describes policy obligations. The requirement "The platform should support multilingual interfaces" was marked as Portability by the LLM, while the human reviewer marked it as Functional and Usability. These examples show that errors are not random; they often occur when a requirement mixes function, user interaction, and organizational constraints.

Practically, these errors suggest two improvements. First, future annotation guidelines should include more boundary examples for Operability, Legal, Look & Feel, Usability, and Portability. Second, model deployment should incorporate human review for low-confidence or multi-label cases, especially when the predicted label belongs to a low-agreement category.

4.8. Practical Implications and Threats to Validity

This study provides three practical takeaways. First, a silver dataset is an effective solution for training when manual annotation is costly. Second, silver and gold data should not be combined into a single benchmark, as they serve different roles and need to be evaluated separately. Third, threshold tuning should be reported clearly, since it can affect model ranking, particularly for classical baselines.

Several threats to validity remain. The silver annotations depend on the prompt design, the retrieved reference examples, and the selected LLM. The reliability assessment relies on a single human reviewer and therefore cannot capture disagreement among human experts. The Vietnamese data are aligned with the English data, which supports fair comparison but may not fully represent natural Vietnamese requirement writing styles. Finally, while we compare threshold tuning with no tuning, a full ablation study of retrieval versus no retrieval should be included in future work.

4.9. Reproducibility and Data Governance

All stages of the workflow are fully reproducible using our scripts, from annotation to training and evaluation. In the final release, we will provide exact data splits, dependencies, random seeds, and standard identifiers to support consistent reuse by other researchers.

Because software requirements may contain organization-specific information, release and deployment should include permission checks, optional de-identification, and proper access control for raw text, rationales, and intermediate annotation files.

For reproducibility, the release should include the raw input files, the final labeled silver dataset, the 500-row human audit sample, the corresponding LLM labels, the kappa report, the fold split files, and the summarized training results. The training scripts should also record the model name, random seed, fold ID, threshold setting, and whether each score is obtained from TEST-SILVER or TEST-GOLD. This distinction is necessary because silver and gold scores address different evaluation questions.

Data governance is also important. Requirement texts may contain organizational vocabulary, domain-specific terms, or operational details. Therefore, public release should include permission checks, optional de-identification, and clear documentation specifying which artifacts are open, restricted, or reproducible only through scripts.

5. CONCLUSION AND FUTURE WORK

In this paper, we develop and evaluate a bilingual silver dataset for software requirements classification. The results show that the silver corpus supports scalable training and internal model comparison, while the gold benchmark remains necessary for conservative external evaluation. The strongest English model on the gold benchmark is RoBERTa-base, and the strongest Vietnamese model is PhoBERT-base-v2. The results also show that scores on the silver test set are consistently higher than those on the gold benchmark, confirming the need to report both evaluation settings separately.

In future work, we plan to expand the gold benchmark, involve multiple human reviewers, refine boundary examples for Operability and Look & Feel, and run ablation studies that remove retrieval or change the number of retrieved examples. We also intend to release the datasets with complete metadata, split files, annotation scripts, and evaluation scripts so that other researchers can reproduce both silver and gold results.

REFERENCES

[1] L. Zhao, W. Alhoshan, A. Ferrari, K. J. Letsholo, M. A. Ajagbe, E.-V. Chioasca, and R. T. Batista-Navarro, "Natural Language Processing for Requirements Engineering: A Systematic Mapping Study," ACM Comput. Surv., vol. 54, no. 3, Art. 55, pp. 55:1-55:41, 2022, doi: 10.1145/3444689.

[2] S.-C. Necula, F. Dumitriu, and V. Greavu-Serban, "A Systematic Literature Review on Using Natural Language Processing in Software Requirements Engineering," Electronics, vol. 13, no. 11, Art. 2055, 2024, doi: 10.3390/electronics13112055.

[3] T. Hey, J. Keim, A. Koziolek, and W. F. Tichy, "NoRBERT: Transfer Learning for Requirements Classification," in Proc. 28th IEEE Int. Requirements Engineering Conf. (RE), Zurich, Switzerland, 2020, pp. 169-179, doi: 10.1109/RE48521.2020.00028.

[4] B. Or, "Improving Requirements Classification with SMOTE-Tomek Preprocessing," arXiv:2501.06491, 2025, doi: 10.48550/arXiv.2501.06491.

[5] J. T. Almonte, S. A. Boominathan, and N. Nascimento, "Automated Non-Functional Requirements Generation in Software Engineering with Large Language Models: A Comparative Study," arXiv:2503.15248, 2025, doi: 10.48550/arXiv.2503.15248.

[6] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, "The Detection and Classification of Non-Functional Requirements with Application to Early Aspects," in Proc. 14th IEEE Int. Requirements Engineering Conf. (RE'06), Minneapolis, MN, USA, 2006, pp. 36-45, doi: 10.1109/RE.2006.65.

[7] G. Boetticher, T. Menzies, and T. Ostrand, PROMISE Repository of Empirical Software Engineering Data. Morgantown, WV, USA: Dept. Comput. Sci., West Virginia Univ., 2007.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv:1907.11692, 2019, doi: 10.48550/arXiv.1907.11692.

[9] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: Pre-trained Language Models for Vietnamese," in Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 2020, pp. 1037-1042, doi: 10.18653/v1/2020.findings-emnlp.92.

[10] P. He, J. Gao, and W. Chen, "DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing," arXiv:2111.09543, 2021, doi: 10.48550/arXiv.2111.09543.

[11] Q. Motger and X. Franch, "Automated Requirements Relation Extraction," in A. Ferrari and G. Ginde, Eds., Handbook on Natural Language Processing for Requirements Engineering. Cham, Switzerland: Springer, 2025, doi: 10.1007/978-3-031-73143-3_7.

[12] M. Vierlboeck, R. Nilchiani, and M. Blackburn, "Natural Language Processing to Assess Structure and Complexity of System Requirements," Syst. Eng., vol. 28, no. 1, pp. 100-109, 2025, doi: 10.1002/sys.21784.

[13] J. Cohen, "A Coefficient of Agreement for Nominal Scales," Educ. Psychol. Meas., vol. 20, no. 1, pp. 37-46, 1960.

[14] K. Sechidis, G. Tsoumakas, and I. Vlahavas, "On the Stratification of Multi-Label Data," in Machine Learning and Knowledge Discovery in Databases, Berlin, Germany: Springer, 2011, pp. 145-158.

[15] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining Multi-label Data," in Data Mining and Knowledge Discovery Handbook, 2nd ed., O. Maimon and L. Rokach, Eds. Boston, MA, USA: Springer, 2010, pp. 667-685, doi: 10.1007/978-0-387-09823-4_34.

[16] M.-L. Zhang and Z.-H. Zhou, "A Review on Multi-Label Learning Algorithms," IEEE Trans. Knowl. Data Eng., vol. 26, no. 8, pp. 1819-1837, 2014, doi: 10.1109/TKDE.2013.39.

[17] A. Ferrari, G. O. Spagnolo, and S. Gnesi, "PURE: A Dataset of Public Requirements Documents," in Proc. IEEE 25th Int. Requirements Engineering Conf. (RE), Lisbon, Portugal, 2017, pp. 502-505, doi: 10.1109/RE.2017.29.