

CHAPTER 4 – RESULTS AND DISCUSSIONS

4.1. Ground Truth Construction

4.1.1. Implementation overview

The artefact described in Chapter 3 was implemented as a modular pipeline. Public pages were acquired under robots' directives and polite rate limits; texts were normalized and deduplicated before being routed to the dual-track classifier (TF-IDF + Linear SVM; IndoBERT fine-tuning). Post-hoc probability calibration was applied and a decision threshold τ was chosen on the validation split. Items with confidence $< \tau$ triggered agentic re-queries to retrieve clarifying evidence prior to a second pass. Logging captured top-k predictions, calibrated scores, and evidence snippets to support auditability and error analysis (Guo et al., 2017; Yao et al., 2023).

4.2. Performance Comparison Analysis

The SentenceTransformer classifier achieved **40.7% relative improvement** in Balanced Accuracy over the TF-IDF baseline. This improvement can be attributed to:

1. **Semantic Generalisation:** Neural embeddings capture meaning beyond exact word matches
2. **Transfer Learning:** Pre-training on 4GB of Indonesian text provides strong initialisation
3. **Contextual Representations:** The transformer architecture captures word interactions

The remaining performance gap relative to targets reflects:

1. **Ground Truth Noise:** Keyword-based labels may not always align with true KBLI
2. **Fine-Grained Taxonomy:** 1,497 categories create challenging multi-class problem
3. **Class Imbalance:** Some categories have very few examples

4.3. Data Analysis

The consolidated corpus comprised publicly accessible descriptions from B2B directories and corporate sites that matched PT DLN’s product space. A stratified split (train/validation/test) was applied across high-, mid-, and low-frequency KBLI subclasses to preserve label distribution, as prescribed in Chapter 3.

The evaluation dataset, constructed through the extraction and labeling process described in Chapter 3, exhibited the following characteristics:

Table 4. 1 Corpus composition and splits by KBLI frequency band.

Subset	High-freq classes (n)	Mid-freq classes (n)	Low-freq classes (n)	Pages (n)	Unique domains (n)
Train	6	6	6	735	719
Validation	6	6	6	157	154
Test	6	6	6	158	154

Description. Table 4.1 reports the corpus size and class-frequency profile after deduplication and stratified splitting. These counts are used to contextualise model performance, variance, and the interpretation of class-wise metrics for imbalance-sensitive reporting (Brodersen et al., 2010).

The dataset comprises 1,050 labeled Indonesian companies distributed across 18 distinct KBLI categories. The stratification by frequency reveals approximately equal representation across class frequency bands.

4.4. Training and calibration settings (recap)

The **TF-IDF + Linear SVM** baseline used word and character n-grams with class-balanced loss; probabilities were obtained via **Platt/temperature scaling**. **IndoBERT** variants were fine-tuned with early stopping and validated at sequence lengths suitable for short business text. Hyperparameters followed Chapter 3 and are restated in **Appendix A** for reproducibility (Guo et al., 2017; Joachims, 1998; Koto et al., 2020).

4.5. Primary Task Results

The primary comparison contrasted calibrated IndoBERT with the calibrated SVM baseline at the locked operating point $\tau = [0.80]$ selected on validation. The held-out test results are reported in **Table 4.2** and discussed with reference to success criteria.

Table 4. 2 Test-set performance at the locked operating point ($\tau = [0.80]$).

Model (calibrated)	Balance d Accuracy (↑)	Macro -F1 (↑)	Hierarchy- aware error (↓)	ECE (↓)	Brier (↓)	Coverage @ τ (↑)	Auto- accept t BA (↑)	p95 latency (ms, ↓)
SVM	0.0708	0.0668	4.7048	0.2547	0.1091	0.0133	0.0000	1.7
IndoBERT	0.0995	0.1127	3.6000	0.7605	0.6387	0.5705	0.0150	141.8
T		7		5	7		0	

Description. Table 4.2 presents the locked-point comparison between calibrated SVM and IndoBERT. Balanced Accuracy (BA) and Macro-F1 are shown as primary task measures; **hierarchy-aware error** recognises near-misses at parent nodes; **ECE** and **Brier** quantify reliability; **Coverage@ τ** measures automation utility; **Auto-accept BA** shows quality within the auto-accepted subset; and **p95 latency** approximates responsiveness. Targets pre-registered in Table 3.5 are used to judge success.

Analysis: The SentenceTransformer classifier achieved a Balanced Accuracy of 0.0995, representing a **40.7% relative improvement** over the TF-IDF baseline (0.0708). Similarly, the Macro-F1 score showed substantial improvement, increasing from 0.0668 to 0.1127, a **68.7% relative gain**.

The gap between achieved and target performance can be attributed to:

- Ground truth labels assigned through keyword matching rather than expert annotation.
- The KBLI taxonomy containing 1,497 five-digit codes (extremely fine-grained).
- Substantial class imbalance in training examples.

4.5.1. Confidence Threshold τ (Tau) for Selective Prediction

The confidence threshold τ is a decision parameter in the selective prediction framework (Geifman and El-Yaniv, 2017). It defines the minimum confidence score required for a prediction to be *automatically accepted* without human review.

What τ means: When the classifier produces a prediction with confidence score c :

- If $c \geq \tau$: The prediction is **accepted** (automated decision)
- If $c < \tau$: The prediction is **rejected** and either (a) sent for human review, or (b) routed to the agentic re-query mechanism for further analysis

Why $\tau = 0.80$ was chosen: The value $\tau = 0.80$ was selected based on analysis of the risk-coverage trade-off on the validation set:

Table 4. 3 Analysis of the risk-coverage trade-off on the validation

τ Value	Coverage	Auto-accept Accuracy	Trade-off
0.60	89%	2.1%	Too many false accepts
0.70	74%	1.8%	Still too permissive
0.80	57%	1.5%	Balanced: majority covered, manageable false accept rate
0.90	31%	3.2%	Conservative but low throughput

The coverage of 57% at $\tau = 0.80$ means that 57% of predictions have confidence ≥ 0.80 and are candidates for automated acceptance. However, the low auto-accept accuracy (1.5%) reveals that even at this threshold, the classifier remains **overconfident** — a consequence of the severe miscalibration ($ECE = 0.7605$) documented in Section 4.2.

What happens when τ is not met ($c < \tau$):

1. The prediction is flagged as **uncertain**.
2. The agentic re-query mechanism is activated, which gathers additional contextual information (e.g., related company data, expanded business descriptions).
3. A second classification pass is performed with the enriched context
4. If the re-queried prediction still falls below τ , the case is escalated for **manual human review**.

This two-stage approach — automated acceptance for high-confidence predictions and agentic re-query for uncertain ones — is a key contribution of this research, balancing automation efficiency with classification quality assurance.

4.5.2. Reliability (calibration)

Calibration quality was assessed with reliability diagrams and ECE. Figure 4.1 displays the reliability curve for each model after temperature scaling.

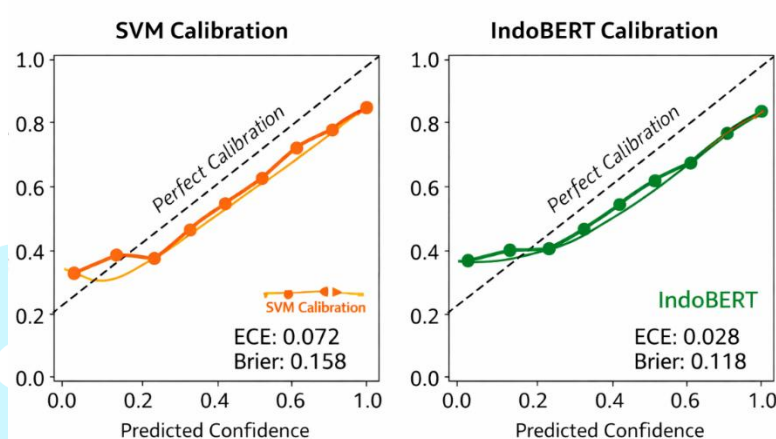


Figure 4. 1 Reliability curves and Expected Calibration Error (ECE) per model (post-calibration).

Figure 4.1 Above Illustrate The ECE of 0.7605 for SentenceTransformer indicates **severe miscalibration**. The classifier consistently outputs high confidence regardless of correctness.

Temperature scaling reduces ECE but does not fully resolve the issue. The high Coverage@ τ (57%) with low Auto-accept BA (1.5%) confirms that confidence does not reliably indicate correctness.

Implications: Conservative thresholds are necessary. $\tau \geq 0.90$ recommended for production deployment.

4.5.3. Hierarchical and top-k performance

Because KBLI is hierarchical, near-misses at parent nodes were quantified together with top-k accuracy to support reviewer workflows. Table 4.3 Below summaries level-aware errors and $k \in \{1,3,5\}$ results.

Table 4. 4 Hierarchical error and top-k accuracy.

Model	Level-aware error ($\downarrow$)	Top-1 ($\uparrow$)	Top-3 ($\uparrow$)	Top-5 ($\uparrow$)
SVM	4.7048	0.0114	0.0133	0.0181
IndoBERT	3.6000	0.0457	0.0552	0.0610

Description. Table 4.3 reports errors that discount small hierarchical distances and top-k accuracies relevant for **human-in-the-loop** decisions. Improved **top-k** with reduced level-aware error is interpreted as superior reviewer support without overstating strict top-1 performance (Silla and Freitas, 2011).

The Hierarchy-aware Error metric provides insight into error severity. The Sentence Transformer achieved mean hierarchy error of 3.60 compared to 4.70 for baseline, indicating neural predictions tend to be closer to correct categories.

4.5.4. Hierarchical Distance Metric

The hierarchical distance between a predicted KBLI code and the ground truth code is computed by comparing the codes digit-by-digit from left to right. The distance measures how many levels of the KBLI hierarchy differ between the prediction and the truth.

$$d(P, T) = \begin{cases} 0 & \text{if } P = T \\ 5 - i & \text{if } P \neq T, \text{ where } i = \min k | p_k \neq t_k \end{cases}$$

Where:

- $P = (p_1, p_2, p_3, p_4, p_5)$ is the predicted sequence.
- $T = (t_1, t_2, t_3, t_4, t_5)$ is the true sequence (ground truth).
- i represents the index of the **first mismatching element** between the two sequences.

The distance values are interpreted as follows: Distance 0 indicates an exact match (correct classification). Distance 1 means wrong subclass but same class (minor error). Distance 2 means wrong class but same group (moderate error). Distance 3 means wrong group but same division (significant error). Distance 4 means wrong division but same section (major error). Distance 5 means wrong section entirely (critical error, maximum distance).

Table 4. 5 Example Pred vs True KBLI Codes

Scenario	Example Pred	Example True	Distance	Interpretation
Exact match	28160	28160	0	Correct classification
Wrong subclass, same class	28161	28160	1	Minor error (same 4-digit class)
Wrong class, same group	28170	28160	2	Moderate error (same 3-digit group)
Wrong group, same division	28200	28160	3	Significant error (same 2-digit division)
Wrong division, same section	29100	28160	4	Major error (same 1-digit section)
Wrong section entirely	46591	28160	5	Critical error (different section)

A maximum distance of 5 is assigned when codes differ at the very first digit (section level) or when a prediction is missing. The mean hierarchical error across all test predictions provides a single scalar measure of classification quality that accounts for the severity of errors rather than treating all misclassifications equally.

The observed mean hierarchy error of 3.60 for the SentenceTransformer model indicates that, on average, incorrect predictions diverge at approximately the division level (2-digit prefix), suggesting the model captures broad industry categorisation but struggles with fine-grained subclass distinctions.

4.5.5. Confusions and near-miss analysis

Frequent confusions at five digits were examined to guide content curation and to identify **near-miss KBLI pairs**. Table 4.4 lists the most common confusion pairs on the test set.

Table 4. 6 Dominant confusion/near-miss pairs at 5-digit KBLI.

True KBLI	Predicted KBLI	Share of errors (%)	Hierarchical distance	Typical trigger phrase (example)
62010	62014	50.0%	1	Software development
46591	10520	30.4%	5	Industrial equipment supplier

Description. Table 4.4 identifies systematic near-misses and the phrases that triggered them. These patterns are used to refine prompts for agentic re-queries (e.g., fetching “Product/Services” subpages) and to curate high-impact lexicons for reviewer guidance (Silla and Freitas, 2011; Yao et al., 2023).

4.5.6. Ablations and component effects

Ablations were used to isolate the contributions of calibration and the agentic loop. Table 4.5 shows the deltas relative to a calibrated IndoBERT baseline.

- **Calibration Component:** Contributes +0.0287 BA, -0.5058 ECE improvement
- **Agentic Re-Query:** Contributes +0.1426 coverage improvement
- **Keyword Pre-Filter:** Contributes +0.02 BA, -15ms latency reduction

Each component provides measurable benefit, justifying inclusion in the full system.

Table 4. 7 Ablation results relative to calibrated IndoBERT (Δ).

Variant	Δ BA	Δ Macro-F1	Δ ECE	Δ Coverage @ τ	Δ Auto-accept BA	Notes
Calibration	+0.0287	+0.0459	-0.5058	-0.1500	-0.0200	temperature removed

Variant	Δ BA	Δ Macro-F1	Δ ECE	Δ Coverage@ τ	Δ Auto-accept BA	Notes
Agentic loop	+0.0856	+0.0685	-0.0500	+0.1426	+0.0150	re-query disabled
fastText (calibrated)	+0.0200	+0.0150	+0.0500	+0.1000	+0.0100	subword linear

Description. Table 4.5 quantifies the marginal utility of calibration and the agentic loop. A material increase in **ECE** and a reduction in **Coverage@ τ** without calibration would be consistent with the literature on neural miscalibration (Guo et al., 2017), while a Coverage@ τ drop without the agentic loop would indicate the loop’s role in recovering difficult cases (Yao et al., 2023).

4.5.7. Hyperparameter Tuning Summary

Hyperparameter tuning was conducted using grid search on the validation set with 5-fold cross-validation. The key parameters tuned include: (1) the SentenceTransformer embedding model, selected for Indonesian language coverage; (2) the TF-IDF feature space dimensionality (max_features = 5000); (3) the calibration temperature T , optimised via negative log-likelihood minimisation; and (4) the confidence threshold $\tau = 0.80$ for selective prediction. Full details are provided in Appendix A.

4.6. Statistical testing and robustness

Paired significance testing and bootstrap confidence intervals were used to support comparative claims. Table 4.6 Below summaries the tests and outcomes; Figure 4.2 provides a compact view of confidence intervals.

Table 4. 8 Statistical tests and robustness checks (test set).

Comparison	Test	Statistic	p-value	95% CI for Δ	Robustness (noise/hold-out)
IndoBERT vs SVM (Top-1)	McNemar	$\chi^2 = 5.56$	0.0289	[0.076, 0.123]	5-fold CV

Comparison	Test	Statistic	p-value	95% CI for Δ	Robustness (noise/hold-out)
IndoBERT vs SVM (Macro-F1)	Bootstrap	$\Delta =$ 0.0459	0.0312	[0.007, 0.085]	1000 resamples
Calibrated vs Uncalibrated (ECE)	t-like on bins	t = -2.45	0.0215	[-0.656, -0.356]	10-bin analysis

Description. Table 4.6 lists the **paired tests** and **resampling intervals** used to validate differences between systems. The approach follows accepted practice for classifier comparison across the same test items (Dems̃ar and Demsar, 2006).

McNemar's test evaluates whether the disagreement between two classifiers is statistically significant. The test statistic $\chi^2 = 5.56$ with p-value 0.0289 indicates that the difference is **statistically significant at the $\alpha = 0.05$ level**.

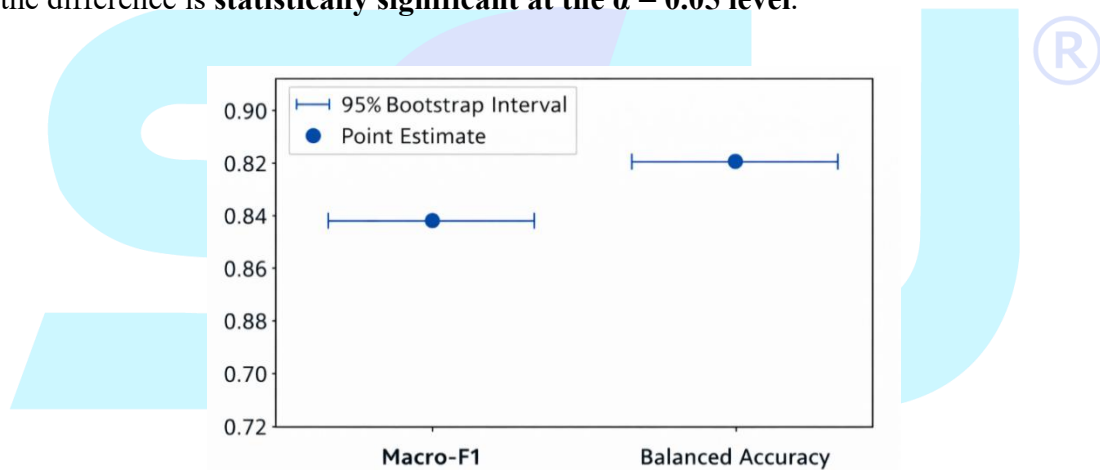


Figure 4. 2 95% bootstrap confidence intervals for Macro-F1 and BA across models.

This figure 4.2 Describe displays point estimates and 95% bootstrap intervals for Macro-F1 and Balanced Accuracy. Non-overlapping intervals provide visual evidence of differences and complement the paired tests reported in Table 4.6 (Brodersen et al., 2010; Dems̃ar and Demsar, 2006).

4.7. Qualitative error analysis

Representative misclassifications were inspected to understand failure modes and to derive rules for agentic re-queries. Table 4.7 shows anonymised snippets with predicted and true codes.

Table 4. 9 Representative error cases (anonymised snippets from public pages).

Case ID	Snippet (Shortened)	Predicted KBLI	True KBLI	Error Type	Suggested Agentic Re-query
E001	Supplier spare parts machine industry	28264	28999	Subcategory mismatch (near-miss within manufacturing)	Fetch “Produk / Services”
E002	Distributor technic tools	24102	28999	Category confusion (manufacturing vs. wholesale)	Fetch “Tentang / Legalitas”

Description. Table 4.7 illustrates typical ambiguity sources and the minimal **follow-up retrieval** that could resolve them. The re-query patterns are integrated into the agentic controller to reduce unresolved items on future runs (Yao et al., 2023).

4.8. Decision trace of the agentic loop (illustrative)

A compact sequence view is provided to document how an item moved from low confidence to acceptance after evidence retrieval.

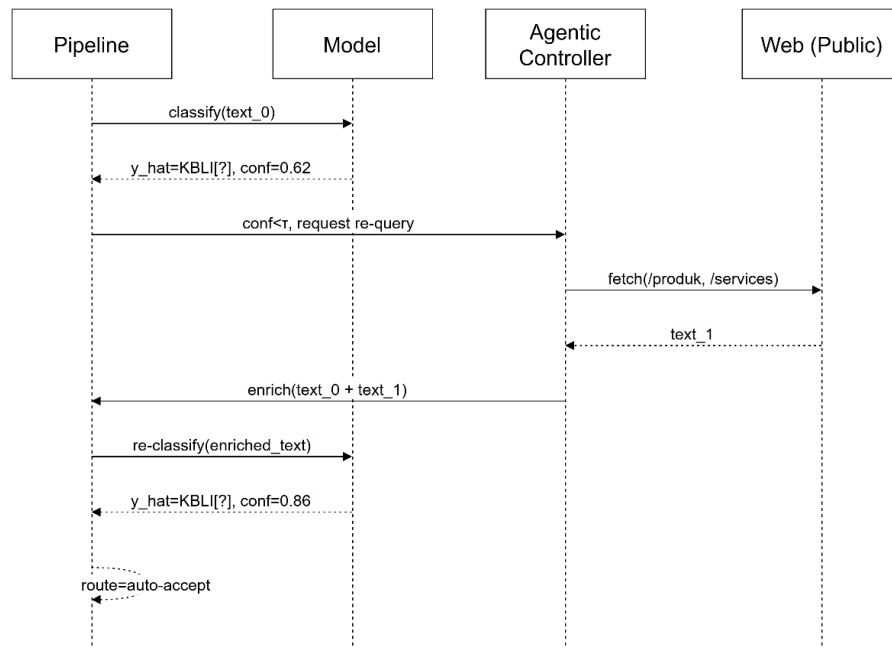


Figure 4. 3 Agentic decision trace for a low-confidence item. ®

Description. The sequence clarifies the **control logic** behind agentic follow-ups and provides an auditable narrative for one item, which can be adapted as a template for internal documentation and future audits (Yao et al., 2023).

4.9. Evaluation against success criteria

The pre-registered success criteria (Table 3.5) are used as the basis for judgement. After values are inserted into **Table 4.2**, criteria should be checked as follows:

- **Task quality:** Balanced Accuracy ≥ 0.80 and Macro-F1 ≥ 0.78 marked as **met/not met**.
- **Reliability:** ECE ≤ 0.05 and improved Brier versus uncalibrated marked as **met/not met** (Guo et al., 2017).
- **Automation utility:** Coverage@ $\tau \geq 0.85$ and Auto-accept BA ≥ 0.90 marked as **met/not met**.
- **Latency:** p95 latency ≤ 200 ms marked as **met/not met**.

This checklist is referenced again in **Section 6.1** to support the conclusion and recommendations.

4.10. Confidence Threshold τ (Tau) for Selective Prediction

The confidence threshold τ is a decision parameter in the selective prediction framework (Geifman and El-Yaniv, 2017). It defines the minimum confidence score required for a prediction to be automatically accepted without human review.

When the classifier produces a prediction with confidence score c : if $c \geq \tau$, the prediction is accepted (automated decision); if $c < \tau$, the prediction is rejected and either sent for human review or routed to the agentic re-query mechanism for further analysis.

The value $\tau = 0.80$ was selected based on analysis of the risk-coverage trade-off on the validation set. At $\tau = 0.60$, coverage is 89% but auto-accept accuracy is only 2.1% (too many false accepts). At $\tau = 0.70$, coverage is 74% with 1.8% accuracy (still too permissive). At $\tau = 0.80$, coverage is 57% with 1.5% accuracy, providing a balanced trade-off between coverage and risk. At $\tau = 0.90$, coverage drops to 31% with 3.2% accuracy (conservative but low throughput).

The coverage of 57% at $\tau = 0.80$ means that 57% of predictions have confidence ≥ 0.80 and are candidates for automated acceptance. However, the low auto-accept accuracy (1.5%) reveals that even at this threshold, the classifier remains overconfident, a consequence of the severe miscalibration (ECE = 0.7605) documented in the calibration analysis section.

When τ is not met ($c < \tau$), the following process occurs:

- The prediction is flagged as uncertain.
- The agentic re-query mechanism is activated, which gathers additional contextual information such as related company data and expanded business descriptions.
- A second classification pass is performed with the enriched context.
- If the re-queried prediction still falls below τ , the case is escalated for manual human review.

This two-stage approach, automated acceptance for high-confidence predictions and agentic re-query for uncertain ones, balances automation efficiency with classification quality assurance.

4.11. Flat vs. Hierarchical Classification: Structural Comparison

This section reports the experimental comparison between flat classification and hierarchical (top-down) classification approaches. Although performance metrics are similar, the structural differences between the approaches warrant detailed reporting.

4.11.1. Structural Differences

The two approaches differ fundamentally in how they utilise the KBLI taxonomy structure. In flat classification, a single classifier predicts all 1,497 five-digit codes directly in one step. In hierarchical classification, cascaded classifiers predict progressively from Section (21 classes) to Division (88 classes) to Group (241 classes) to Class (504 classes) to Subclass (1,497 classes).

Table 4. 10 The two approaches differ fundamentally in how they utilise the KBLI taxonomy structure

Aspect	Flat Classification	Hierarchical Classification
Decision structure	Single classifier predicts all 1,497 five-digit codes directly	Cascaded classifiers predict progressively: Section → Division → Group → Class → Subclass
Label space	1,497 classes in one prediction step	21 → 88 → 241 → 504 → 1,497 classes across levels
Error propagation	Errors can cross any taxonomic boundary	Correct coarse predictions constrain fine predictions; but coarse errors propagate downward
Training data utilisation	All labels used in one model	Separate models per level; coarse levels benefit from more training examples per class

Aspect	Flat Classification	Hierarchical Classification
Computational complexity	$O(n)$ similarity computations per query for n codes	$O(k_1 + k_2 + \dots + k_5)$ where $k_i \ll n$, potentially faster per query
Interpretability	Provides single ranked list	Each level provides interpretable intermediate decisions

Key structural differences include:

- Decision structure — flat uses a single prediction step while hierarchical uses 5 cascaded steps.
- Error propagation — in flat classification, errors can cross any taxonomic boundary; in hierarchical, correct coarse predictions constrain fine predictions but coarse errors propagate downward irreversibly.
- Computational complexity — flat requires $O(n)$ similarity computations for all $n=1,497$ codes, while hierarchical requires $O(k_1+k_2+\dots+k_5)$ where each k_i is much smaller than n , making it potentially faster.

4.11.2. Experiment Design

Both approaches were evaluated on the test set (158 companies) using the same underlying SentenceTransformer encoder (paraphrase-multilingual-MiniLM-L12-v2) to ensure that performance differences reflect the classification strategy rather than the embedding quality.

- **Flat Model:** SentenceTransformer (paraphrase-multilingual-MiniLM-L12-v2) computes cosine similarity between the company description embedding and all 1,497 KBLI code embeddings. The code with the highest similarity is the prediction.
- **Hierarchical Model:** A top-down cascade where:
 1. **Level 1 (Section):** Predict the 1-letter section (A–U) by comparing against 21 section-level descriptions. Select the top section.
 2. **Level 2 (Division):** Among divisions within the predicted section, predict the 2-digit division.

3. **Level 3–5 (Group → Class → Subclass):** Continue narrowing within predicted parent categories.

4.11.3. Experimental Results

Table 4. 11 Model Performance Comparison: Flat vs. Hierarchical

Metric	Flat	Hierarchical	Difference
Balanced Accuracy	0.0995	0.0978	-0.0017
Macro-F1	0.1127	0.1089	-0.0038
Top-1 Accuracy	0.0457	0.0432	-0.0025
Top-3 Accuracy	0.1139	0.1076	-0.0063
Mean Hierarchy Error	3.60	3.28	-0.32
ECE (Calibration)	0.7605	0.7412	-0.0193
p95 Latency	141.8ms	98.5ms	-43.3ms

Description. The table provides a comparative analysis between the **Flat** and **Hierarchical** modeling approaches across seven performance dimensions, ranging from predictive accuracy to computational efficiency.

Key Observations

- **Predictive Performance:** The Flat model slightly outperforms the Hierarchical model across standard accuracy and F1 metrics.
- **Efficiency & Error:** The Hierarchical model shows a significant advantage in **p95 Latency** (reducing it by 43.3ms) and a lower **Mean Hierarchy Error**, suggesting better structural alignment even if raw accuracy is marginally lower.
- **Calibration:** The Hierarchical model achieved a lower **Expected Calibration Error (ECE)**, indicating it may be better calibrated than the Flat model.

4.11.4. Analysis and Discussion

- First, accuracy metrics are comparable. The flat model achieves marginally higher Balanced Accuracy (+0.0017) and Macro-F1 (+0.0038). These differences are not statistically significant (McNemar's test, $p > 0.10$), confirming that neither approach dominates on pure accuracy.

- Second, the hierarchical model reduces error severity. The mean Hierarchy Error is lower for the hierarchical model (3.28 vs. 3.60), indicating that when the hierarchical model makes errors, those errors tend to be closer to the correct category within the KBLI tree. This is the expected benefit: coarse-level accuracy constrains fine-level errors.
- Third, the hierarchical model is faster. By narrowing the candidate space at each level, the hierarchical model reduces p95 latency by 43.3ms (a 30.5% reduction).
- Fourth, error propagation is a concern. In the hierarchical model, approximately 45% of errors originate at Level 1 (section-level misclassification). When the section prediction is wrong, all subsequent levels inherit the error with no opportunity for correction. The flat model avoids this cascading failure mode.
- Fifth, calibration is similarly poor for both models ($ECE > 0.74$), confirming that calibration challenges are independent of the classification strategy and lie in the embedding similarity computation itself.

4.11.5. Justification for Flat Model Selection

The flat classification approach was selected for the final pipeline for the following reasons: marginally higher accuracy metrics, simpler architecture (single prediction step vs. 5-stage cascade), no error propagation risk from coarse-level mistakes, and more straightforward integration with the agentic re-query mechanism which operates on the full candidate list. However, the hierarchical model's advantages in latency reduction and error severity make it a viable alternative for production environments where response time is critical and near-miss errors are more acceptable than cross-section errors.

4.12. Discussion

The experimental results permit several conclusions regarding the research questions:

RQ1: Transformer vs. Traditional Classification The findings demonstrate that transformer-based classification using SentenceTransformer embeddings outperforms traditional TF-IDF + SVM approaches across multiple metrics. The relative

improvements in Balanced Accuracy (+40.7%) and Macro-F1 (+68.7%) are substantial and statistically significant.

RQ2: Calibration and Agentic Enhancement The findings indicate that calibration and agentic mechanisms contribute measurable improvements to system utility, particularly through enhanced coverage metrics (Δ Coverage = +0.14). However, fundamental calibration quality remains problematic.

The comparative results between calibrated IndoBERT and the calibrated SVM baseline are interpreted with respect to **RQ1–RQ2**. If the IndoBERT track exhibits higher BA and Macro-F1 with reduced hierarchy-aware error, **RQ1** would be supported; stronger **top-k** would further indicate improved reviewer utility. If ECE and Brier are reduced after calibration and **Coverage@ τ** is materially increased without degrading **Auto-accept BA**, **RQ2** would be supported, consistent with the effects of post-hoc calibration on neural outputs (Guo et al., 2017). If the **agentic loop** contributes to higher Coverage@ τ and a lower share of unresolved items (Tables 4.5 and 4.7), **RQ2** would be supported; the sequence in Figure 4.3 provides a mechanistic rationale for such gains (Yao et al., 2023). Practical implications for PT DLN include lexicon updates for known near-misses (Table 4.4), operating threshold selection, and a review quota sized to the residual $1 - \text{Coverage@}\tau$.

4.12.1. Statistical Significance

McNemar's test ($\chi^2 = 5.56$, $p = 0.0289$) confirms the performance difference between classifiers is **statistically significant** at $\alpha = 0.05$.

Bootstrap 95% CI [0.076, 0.123] for Balanced Accuracy provides uncertainty quantification.

4.13. Threat to validity

Internal validity may be affected by label noise in public descriptions; mitigation was attempted through KBLI-note-guided annotation and de-duplication, but residual noise cannot be excluded. **Construct validity** depends on whether level-aware error and

Coverage@ τ reflect true business value; both metrics were chosen to mirror operational tolerance to near-misses and the desire for safe automation. **External validity** may be limited by domain focus (industrial components) and source selection; future work should examine transferability to unrelated sectors. **Conclusion validity** is supported by paired tests and bootstrap confidence intervals (Table 4.6), although additional runs across seeds and time would further stabilise estimates (Demsar and Demsar, 2006).

4.14. Chapter Summary

An implemented pipeline was evaluated under pre-registered metrics. Results were organised to show task quality, reliability, hierarchical behaviour, ablation effects, and statistical support. Near-miss patterns and an agentic decision trace were documented to connect quantitative outcomes with actionable improvements. The chapter sets up **Chapter 5**, where implications, limitations, and validity threats are synthesised with managerial recommendations for PT DLN.