TextGuard: Risk & Compliance Flagging for IDB Project Documents

Anonymous Authors[1]

Abstract

We present **TextGuard**, a chunk-level transformer pipeline for triaging risk and compliance signals in long Inter-American Development Bank (IDB) project PDFs. TextGuard parses documents into *semantically meaningful chunks* (rather than fixed token windows), fine-tunes BERT-based classifiers for *risk* and *compliance*, and aggregates chunk probabilities into document scores via a length-weighted average. Our labeled corpus contains 50 documents segmented into 263 sections and 958 chunks with substantial class imbalance, which makes accuracy alone misleading. On a held-out **chunk-level** test split, our risk model achieves **0.910** accuracy and **0.710** macro-F1, while the tuned compliance model reaches **0.969** accuracy and **0.956** macro-F1. These results highlight the effectiveness of semantic chunking and macro-metric optimization for evidence-based triage in imbalanced, high-stakes document review.

1. Introduction

Development-bank project documents often span dozens of pages and combine governance, safeguards, procurement, implementation plans, and monitoring language. Reviewers must identify passages that signal potential concerns and determine whether text indicates compliance or non-compliance with requirements. Scaling this is hard: actionable evidence is scattered across sections, and domain terminology frequently appears in both problem statements and mitigation narratives, making naive keyword search and overly coarse document-level classification unreliable.

We built TextGuard to tackle **review triage** by prioritizing evidence over black-box labels. Instead of only labeling whole documents, TextGuard predicts at the **chunk level** and surfaces the exact text blocks responsible for predictions. These chunk outputs are then aggregated into document-level scores to rank documents while preserving transparency: reviewers can validate model suggestions by reading the predicted evidence chunks.

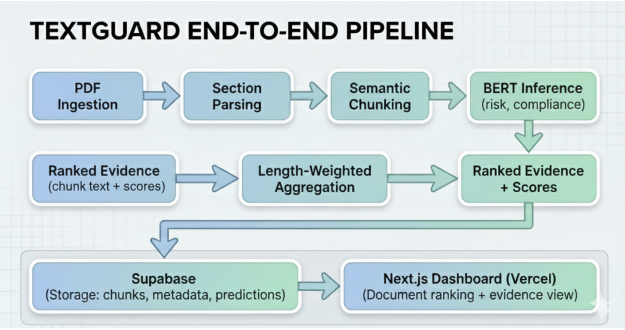

Figure 1. TextGuard end-to-end pipeline: PDF parsing → semantic chunking → chunk-level transformer inference → length-weighted aggregation → ranked evidence for reviewer triage.

Contributions. (1) A PDF→section→semantic-chunk pipeline that produces interpretable training instances and reviewer-readable evidence. (2) Transformer-based chunk classifiers for *risk* and *compliance* evaluated under strong label imbalance using macro metrics and per-class behavior. (3) A document scoring scheme based on length-weighted aggregation that supports ranking and triage while retaining chunk-level provenance.

2. Related Work

Transformers and domain adaptation. Our chunk classifiers fine-tune BERT-based encoders for sequence classification (Devlin et al., 2019). Prior work shows that continuing pretraining on in-domain corpora can significantly improve downstream performance in specialized settings (Gururangan et al., 2020). Domain-adapted variants such as LEGAL-BERT demonstrate similar gains for legal/compliance-like text (Chalkidis et al., 2020), motivating our focus on task-specific fine-tuning and robust evaluation.

Long-document modeling and hierarchical approaches. Long documents pose challenges for vanilla transformers due to context-length limits. Sparse-attention and long-context architectures such as Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020) extend usable con-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

text windows. Hierarchical modeling offers an alternative by leveraging document structure, including hierarchical attention networks (Yang et al., 2016) and more recent hierarchical transformer designs (He et al., 2024; Chalkidis et al., 2022). TextGuard follows a pragmatic approach: it preserves interpretability by converting long PDFs into paragraph-scale semantic chunks for standard transformer encoders.

Semantic chunking and segment aggregation. A common strategy for long-text decision making is *segmentation* followed by *aggregation* of segment-level predictions. Segmentation-and-aggregation pipelines have been shown to improve stability and interpretability in legal judgment prediction (Almalki et al., 2025). Recent work also emphasizes semantic chunking and chunk-level evaluation as key ingredients in robust retrieval and reasoning pipelines (Singh et al., 2024). TextGuard adopts this paradigm by producing chunk-level evidence and aggregating probabilities into document-level triage scores.

Evidence-centric decision support and accountability. Rather than returning only document labels, TextGuard surfaces predicted evidence chunks to support human verification. This aligns with rationale-based interpretability approaches (Lei et al., 2016) and broader calls for end-to-end accountability and auditing in deployed ML systems (Raji et al., 2020).

Imbalanced classification. Class imbalance is common in institutional text (where most content is neutral/procedural). Focal loss (Lin et al., 2017) is a widely used approach for emphasizing hard examples; in this work, we report macro-averaged metrics to avoid misleading accuracy under imbalance.

3. Problem Setup

TextGuard operates at the chunk level. Given a document D, we extract a sequence of chunks $\{x_1, \ldots, x_n\}$, where each x_i is a semantically coherent text block associated with a document section.

We tackle two classification tasks:

Risk detection: predict

$$y_i^{(r)} \in \{\texttt{risk}, \texttt{no\_risk}\}$$

for each chunk x_i.

Compliance classification: predict

$$y_i^{(c)} \in \{\texttt{Compliant}, \texttt{Non-Compliant}\}$$

for each chunk x_i (excluding a small number of `Not Applicable` entries from training/evaluation).

Because review is performed at the document level, TextGuard also computes a document score from chunk outputs to rank documents for triage, while retaining chunk-level predictions as evidence.

4. Data and Preprocessing

4.1. Document selection and representativeness

We selected 50 IDB project PDFs spanning multiple sectors and document templates. The corpus is limited in size, and we use chunk-level splits rather than document-level splits as specified in our experimental protocol (Sec. 4.5).Since no public benchmark dataset exists for IDB risk/compliance signals at the chunk level, we constructed our own labeled corpus by operationalizing IDB policy language into annotation guidelines and creating a gold-standard dataset that can support future extensions and evaluation. We discuss the implications for generalization to unseen documents in Sec. 9.

4.2. Document parsing and semantic chunking

We ingest IDB project PDFs and extract structured section text using document layout cues (e.g., headings and section boundaries). Each section is then split into **semantically meaningful chunks** (typically paragraph-scale blocks) rather than fixed-length token windows. This choice improves interpretability: a predicted chunk can be shown directly as evidence to a reviewer without requiring surrounding context to be useful.

4.3. Dataset statistics and labels

Our labeled corpus contains 50 documents segmented into 263 sections and 958 chunks. The dataset is heavily imbalanced (6.4:1 ratio of risk to no risk chunks), which is typical in institutional documents where most content is procedural or mitigation-focused. For risk detection, we have 828 risk chunks and only 130 no risk chunks. Compliance labels include `Compliant`, `Non-Compliant`, and a small number of `Not Applicable` cases (5 chunks in the full corpus). Reported test metrics are computed on the classes present in the held-out split; in our test split this corresponds to Compliant vs Non-Compliant.

4.4. Sector-specific policy taxonomy (risk types)

Beyond binary risk/no risk, we annotate risk-bearing chunks with a sector-specific policy label drawn from a curated taxonomy. Each project sector is associated with a fixed set of policy topics (e.g., TRANSPORT includes `transport safety`, `freight and goods movement`, `climate resilience and adaptation`). This makes predictions more actionable:

flagged evidence can be grouped and summarized by policy topic for reviewer follow-up.

In our taxonomy we cover **17 sectors** with **177 unique policy labels** (avg. 10.4 labels/sector; min 8, max 18). Sectors span major IDB areas (e.g., *Education*, *Energy*, *Transport*, *Health*, *Financial Markets*, *Water and Sanitation*, and *Urban Development and Housing*). The full taxonomy is available upon request.

4.5. Splits and leakage considerations

We evaluate TextGuard using a chunk-level random split of the labeled chunks into train/validation/test partitions (80/10/10) with a fixed random seed (42). Each chunk inherits metadata from its source PDF (e.g., document id), but splitting is performed at the chunk level rather than grouping by document.

This choice can admit within-document leakage: chunks originating from the same PDF may appear in both train and test, allowing the model to benefit from repeated phrasing, named entities, or template structure shared across sections. Therefore, results in this draft should be interpreted as measuring chunk-level classification performance under this split, rather than document-level generalization to wholly unseen PDFs. We discuss document-level evaluation as future work in Sec. 9.

5. Method

5.1. Chunk-level transformer classifiers

We fine-tune BERT-style sequence classifiers (`bert-base-uncased`) for both tasks. Each chunk x_i is tokenized and encoded by a transformer encoder, and a classification head maps the pooled representation to class logits. The model outputs a probability p_i for the target class (e.g., risk) and is trained with cross-entropy loss using standard transformer fine-tuning procedures (e.g., AdamW optimization and validation-based model selection).

5.2. Evidence-first outputs

TextGuard is designed to be evidence-centric: it stores and returns chunk-level predictions alongside the corresponding chunk text. In the reviewer workflow, high-scoring chunks provide inspectable rationales for a document's triage score, enabling quick verification and reducing over-reliance on an opaque document-level label.

5.3. Length-weighted document scoring

To obtain a document-level triage score while reflecting the amount of supporting evidence, we aggregate chunk probabilities using a length-weighted average:

$$S_{\text{doc}} = \frac{\sum_{i=1}^{n} w_i \, p_i}{\sum_{i=1}^{n} w_i},$$

where w_i is the chunk length (e.g., word count). This down-weights very short chunks so the score better reflects the volume of evidence in the document.

5.4. System implementation

TextGuard stores extracted sections, chunk text, metadata, and model outputs in Supabase. A Next.js (React) frontend deployed on Vercel retrieves results and renders document ranking, aggregate scores, and evidence chunks for review. This separation supports iterative model updates while keeping the reviewer workflow stable.

6. Experiments

6.1. Metrics

Because both tasks are imbalanced, we report accuracy and macro-averaged precision, recall, and F1. Macro-averaged F1 treats classes equally, which better captures minority-class performance than accuracy alone. We additionally use per-class precision/recall/F1 to diagnose failure modes on rare labels (no risk and Non-Compliant).

6.2. Baselines and tuned models

We compare a baseline BERT configuration against a tuned version we developed through iterative refinement (e.g., improved chunking consistency and training/selection adjustments). Since minor preprocessing changes can slightly alter the test set composition across iterations, we interpret results primarily through macro metrics and class-wise behavior rather than accuracy alone.

6.3. Reproducibility details

We fine-tune `bert-base-uncased` using the HuggingFace Trainer. Chunks are tokenized with a maximum sequence length of 256 tokens. For risk detection we use learning rate 2×10^{-5}, batch size 8, weight decay 0.01, and train for 3 epochs. For compliance we use learning rate 2×10^{-5}, batch size 16, weight decay 0.01, and train for 3 epochs. We use a fixed random seed (42) for data splitting.

7. Results and Discussion

Table 1 shows our chunk-level results on the held-out test split. We focus on accuracy and macro-averaged precision/F1 to account for class imbalance (Sec. 6.1). Table 1 summarizes baseline vs. tuned performance for both tasks, while Fig. 2 and Fig. 3 provide confusion matrices to high-

Table 1. Chunk-level held-out performance. Macro metrics are emphasized due to class imbalance.

Task / Model	Acc	Macro-P	Macro-F1
Risk (baseline)	0.8110	0.6480	0.6780
Risk (tuned)	0.9100	0.7510	0.7100
Compliance (baseline)	0.8840	0.8720	0.7280
Compliance (tuned)	0.9688	0.9375	0.9565

Confusion matrix (Compliance)

	Compliant	Non-Compliant
Compliant	72	3
Non-Compliant	0	21

Figure 2. Confusion matrix on the compliance test set.

light class-wise behavior, especially on the minority classes (Non-Compliant and no risk). Finally, we include brief qualitative error analysis (Table 2) to illustrate typical failure modes observed in misclassified chunks.

7.1. Qualitative error analysis

We inspect misclassified test chunks to identify common failure modes, such as risk-related terminology appearing in mitigation contexts (false positives), or neutral administrative language masking risk (false negatives).

7.2. Risk detection under imbalance

The tuned risk classifier improves accuracy from 0.811 to 0.910 and macro-F1 from 0.678 to 0.710, indicating gains that persist beyond majority-class effects. But error analysis shows minority-class behavior is still the bottleneck: no risk is relatively rare (130/958), and the model tends to favor risk when risk-related terminology appears in neutral or mitigation contexts. This pattern is consistent with institutional documents where "risk language" is common even when describing safeguards and controls.

7.3. Compliance classification

The tuned compliance model achieves 0.9688 accuracy and macro-F1 0.9565, with macro-recall 0.98, which shows strong balanced performance despite the class imbalance (Compliant dominates). In a reviewer workflow, high recall is particularly valuable because missing truly Non-Compliant evidence can be more costly than surfacing additional chunks for verification.

Confusion matrix (Risk)

	no_risk	risk
no_risk	6	6
risk	2	82

Figure 3. Confusion matrix on the risk test set.

Table 2. Example misclassifications (held-out chunks).

Task	True→Pred	Chunk excerpt
Risk	`no_risk→risk`	Project is available for the public under a Creative Commons license . . .
Comp	`Compliant→Non-Compliant`	The Environmental and Social . . . Procurement . . .

7.4. Operational implications

TextGuard mitigates model risk by emphasizing evidence: reviewers see the actual flagged chunks and can accept or reject model suggestions. Document-level scoring supports prioritization, while chunk-level provenance prevents the system from becoming a black-box "approval" mechanism.

8. Impact Statement

TextGuard is intended to support rather than replace expert review of development-bank project documents. By ranking documents using predicted risk/compliance signals and surfacing the specific evidence chunks responsible for those scores, the system can reduce time spent on manual scanning and help reviewers focus attention where it is most needed. This could improve consistency and speed up triage when reviewers face large document volumes.

That said, the system isn't without risks. False negatives can be costly if truly Non-Compliant or high-risk evidence is missed, while false positives can waste reviewer time and reduce trust in the tool. Models trained on limited and imbalanced labels may encode dataset-specific biases and generalize poorly to new sectors, countries, or document styles. Finally, automated scores could be misused as a substitute for human judgment in high-stakes settings.

To mitigate these risks, TextGuard is evidence-centric: it always presents chunk text alongside predictions, supports threshold tuning to match institutional risk tolerance, and encourages human verification. Future work will prioritize expanding minority-class coverage, improving calibration and uncertainty reporting, and adding policy-linked explanations to strengthen interpretability and safe deployment.

9. Limitations

Label imbalance and minority-class sensitivity. Both tasks are substantially imbalanced, which can inflate accuracy while masking poor minority-class recall (e.g., no risk and Non-Compliant). Although macro-F1 partially addresses this, the model can still under-detect rare but critical cases. Our next steps include collecting more minority examples and exploring imbalance-aware training (e.g., class-weighted objectives or focal loss), along with calibrated thresholds that reflect reviewer risk tolerance.

Dataset size and domain coverage. The labeled corpus is modest in size and may not represent the full diversity of IDB sectors, countries, and document styles. As a result, performance may degrade under domain shift (new templates, writing styles, or sector-specific vocabulary). Document-level holdout splits reduce leakage, but broader coverage and multi-domain evaluation would strengthen external validity.

Chunk-level modeling limits global context. Our current evaluation uses a chunk-level random split, which can admit within-document leakage when chunks from the same PDF appear in both train and test. As a result, the reported metrics primarily reflect chunk-level classification performance under this split and may overestimate performance on entirely unseen documents. A key next step is document-level splitting by document id to measure generalization to new PDFs without leakage. Chunking improves interpretability and scalability, but it can miss dependencies that span distant sections (e.g., a requirement stated early and evidence discussed later). This may lead to inconsistent chunk predictions for the same underlying issue. Hybrid approaches such as lightweight long-context models, hierarchical aggregation, or retrieval-based re-ranking of relevant context could improve global coherence.

Operational calibration and human factors. In triage settings, the optimal decision threshold depends on whether reviewers prefer higher recall (fewer misses) or higher precision (fewer false alarms). Without calibration and explicit uncertainty reporting, confidence scores may be misinterpreted. TextGuard mitigates this by surfacing evidence chunks, but safe deployment still requires threshold tuning, monitoring, and periodic re-validation.

10. Conclusion

We built TextGuard, an evidence-centric system for triaging risk and compliance signals in long IDB project documents. TextGuard parses documents into semantically meaningful chunks, fine-tunes BERT-based chunk classifiers, and aggregates chunk probabilities into document-level scores via length-weighted averaging while preserving provenance through ranked evidence chunks. On held-out evaluations under strong class imbalance, the tuned risk model achieves 0.910 accuracy and 0.710 macro-F1, and the tuned compliance model reaches 0.9688 accuracy and 0.9565 macro-F1.

The main remaining challenges are minority-class robustness and generalization across document styles. We plan to expand minority-label coverage, improve calibration and uncertainty reporting, and incorporate policy-linked explanations to strengthen interpretability and safe use in real review workflows.

References

Almalki, S., Alsafari, M., and Alotaibi, M. Legal judgment prediction in the saudi arabian commercial court. *Future Internet*, 17(10):439, 2025. doi: 10.3390/fi17100439.

Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., and Androutsopoulos, I. LEGAL-BERT: The muppets straight out of law school. In *Findings of EMNLP*, pp. 2898–2904, 2020.

Chalkidis, I., Dai, X., Fergadiotis, M., Malakasiotis, P., and Elliott, D. An exploration of hierarchical attention transformers for efficient long document classification. *arXiv preprint arXiv:2210.05529*, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, 2019.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 8342–8360, 2020.

He, H., Flicke, M., Buchmann, J., Gurevych, I., and Geiger, A. Hdt: Hierarchical document transformer. In *Conference on Language Modeling (COLM)*, 2024.

Lei, T., Barzilay, R., and Jaakkola, T. Rationalizing neural predictions. In *Proceedings of EMNLP*, pp. 107–117, 2016.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of ICCV*, pp. 2980–2988, 2017.

Raji, I. D., Smart, A., White, R. N., et al. Closing the ai accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT)*, pp. 33–44, 2020.

Singh, I. S., Aggarwal, R., Allahverdiyev, I., Taha, M., Akalin, A., Zhu, K., and O'Brien, S. Chunkrag: Novel llm-chunk filtering method for rag systems. *arXiv preprint arXiv:2410.19572*, 2024.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pp. 1480–1489, 2016.

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., and Ahmed, A. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.