AI for Climate Finance: Agentic Retrieval and Multi-Step Reasoning for Early Warning System Investments

Ario Saeid Vaghefi[1,2*], **Aymane Hachcham**[1*]
Veronica Grasso[2], **Nakiete Msemo**[2], **Chiara Colesanti Senni**[1]
Markus Leippold[1,3]
[1]University of Zurich [2]WMO [3]Swiss Finance Institute (SFI)
{saeid.vaghefi, aymane.hachcham, chiara.colesantisenni, markus.leippold}@df.uzh.ch
{svaghefi, vgrasso, nmsemo}@wmo.int

Abstract

Tracking financial investments in climate adaptation is complex and expertise-intensive, particularly for Early Warning Systems (EWS), where multilateral development bank (MDB) and fund reports lack standardized financial reporting and appear as heterogeneous PDFs with complex tables and inconsistent layouts.

We introduce an agent-based Retrieval-Augmented Generation (RAG) system that uses hybrid retrieval and internal chain-of-thought (CoT) reasoning to extract relevant financial data, classify EWS investments, and allocate budgets with grounding evidence spans. While these components are individually established, our contribution is their integration into a domain-specific workflow tailored to heterogeneous MDB reports and numerically grounded EWS budget allocation. On a manually annotated CREWS Fund corpus, our system outperforms four alternatives (zero-shot classifier, few-shot "zero rule" classifier, fine-tuned transformer-based classifier, and few-shot CoT+ICL classifier) on multi-label classification and budget allocation, achieving 87% accuracy, 89% precision, and 83% recall. We further benchmark against the Gemini 2.5 Flash AI Assistant on an expert-annotated MDB evidence set co-curated with the World Meteorological Organization (WMO), enabling a comparative analysis of glass-box agents versus black-box assistants in transparency and performance. The system is publicly deployed and accessible at https://ews-front.vercel.app/ (see Appendix B for demonstration details and Appendix E for dataset statistics and splits).[1]

1 Introduction

Recent advances in Large Language Models (LLMs) have improved automated analysis of financial documents, yet tracking investments in Early Warning Systems (EWS) remains difficult because Multilateral Development Bank (MDB) and climate-fund reports lack standardized labels, structures, and terminology for EWS-related spending. EWS are central to disaster risk reduction and climate resilience, with the UN's Early Warnings for All (EW4All) initiative targeting universal coverage by 2027, but current reporting practices leave EWS financial flows opaque and hinder efficient allocation of climate-finance resources. We frame this problem as a combined multi-label classification and budget allocation task: the system assigns each text or table snippet to one or more EWS pillars and extracts pillar-level budget allocations with grounding evidence spans, producing a structured JSON output over the five EW4All pillars (see Appendix D for definitions and examples).

Contributions. We present the *EW4All Financial Tracking AI-Assistant*, a glass-box, agent-based Retrieval-Augmented Generation (RAG) system that parses heterogeneous MDB project documents, classifies EWS investments across pillars, and returns numerically grounded, evidence-linked budget allocations. Our key contributions are:

1. A novel agent-based RAG pipeline integrating iterative sub-query generation, hybrid semantic-lexical retrieval, self-validation guardrails, and schema-aware consolidation for climate finance document analysis.
2. A publicly deployed system accessible at https://ews-front.vercel.app/, enabling practitioners to analyze MDB documents in real-time.
3. A comprehensive evaluation on a manually annotated CREWS-Fund corpus where our pipeline achieves 87% accuracy, 89% precision, and 83% recall, outperforming four strong baselines.
4. A comparative study against black-box assis-

[*] Equal Contributions.

[1] We will open-source all code, LLM generations, and human annotations to support further work on AI-assisted climate finance.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 492–511
March 24-29, 2026 ©2026 Association for Computational Linguistics

tants (Gemini 2.5 Flash, OpenAI Assistants) on an expert-annotated MDB evidence set co-curated with WMO.

5. Open-source release of expert-annotated corpus, benchmark dataset, and all prompt designs to catalyze future research.

Implications. By turning unstructured MDB reports into structured, evidence-based EWS investment profiles, our system improves climate-finance transparency, accountability, and decision support for MDBs, funds, and technical partners. The combination of RAG and agentic reasoning yields traceable outputs that support portfolio screening, gap analysis across EWS pillars, and monitoring of progress toward EW4All objectives, and offers a transferable blueprint for AI-assisted analysis of climate adaptation and development finance.

2 Related Work

RAG augments LLMs with external retrieval for knowledge-intensive tasks (Lewis et al., 2020), but static pipelines limit adaptability. Recent *agentic RAG* introduces iterative retrieval and decision-making, improving factuality and multi-step reasoning (Xi et al., 2023; Yao et al., 2023; Guo et al., 2024), while multi-agent variants specialize roles for tasks such as code generation and verification and enhance explainability and human–AI collaboration (Guo et al., 2024; Liu et al., 2024). In parallel, in-context learning (ICL) enables few-shot generalization without fine-tuning (Brown et al., 2020); retrieval-based ICL and reward models optimize demonstration selection (Wang et al., 2024). Chain-of-thought (CoT) prompting improves stepwise reasoning (Wei et al., 2022; Kojima et al., 2022), with self-consistency and active example selection further boosting complex question-answering performance (Wang et al., 2023; Diao et al., 2024).

3 System Overview

MDB project documents possess highly heterogeneous layouts—mixed narrative text, nested tables, multi-column formats, and scattered financial evidence—making conventional retrieval pipelines insufficient for accurate budget extraction. To address this, we developed the *EW4All Financial Tracking AI-Assistant*, an agent-based RAG system that integrates hybrid retrieval with hierarchical reasoning.

As illustrated in Figure 1, our pipeline consists of five integrated stages. First, we process documents using the Docling parser to extract raw text and structural elements, followed by context-augmented chunking where each chunk is enriched with a document-level summary to reduce semantic ambiguity. Second, we employ hybrid retrieval that fuses dense vector search (using OpenAI embeddings) and sparse lexical search (BM25F) via Reciprocal Rank Fusion (RRF) to capture both semantic meaning and exact financial figures.

Third, an LLM Agent orchestrates the reasoning process by generating iterative sub-queries and validating retrieved evidence against coverage thresholds. If the retrieved context is insufficient, the agent triggers a self-healing loop to re-query the database. Finally, the system executes schema-aware consolidation, mapping the extracted evidence to the five EWS pillars and allocating budgets with explicit evidence grounding.

The full technical implementation, including embedding construction, hybrid rank fusion equations, and the agent's control flow, is detailed in Appendix A.

4 System Demonstration

The EW4All Financial Tracking AI-Assistant is publicly deployed and accessible at `https://ews-front.vercel.app/`. This section describes the system's user interface, key features, and demonstration scenarios.

4.1 Interface Overview

The web-based interface provides an intuitive workflow for climate finance analysts:

1. **Document Upload:** Users can upload MDB project documents in PDF format. The system accepts documents from various MDBs and climate funds, handling heterogeneous layouts automatically.
2. **Real-Time Processing:** Upon upload, the system displays processing progress with intermediate reasoning steps, allowing users to observe the agent's sub-query generation and retrieval operations.
3. **Interactive Results:** The classification results are presented with:
   - Pillar-wise budget allocations with confidence scores
   - Clickable evidence spans that highlight source passages in the original PDF
   - A visual distribution chart showing budget allocation across EWS pillars

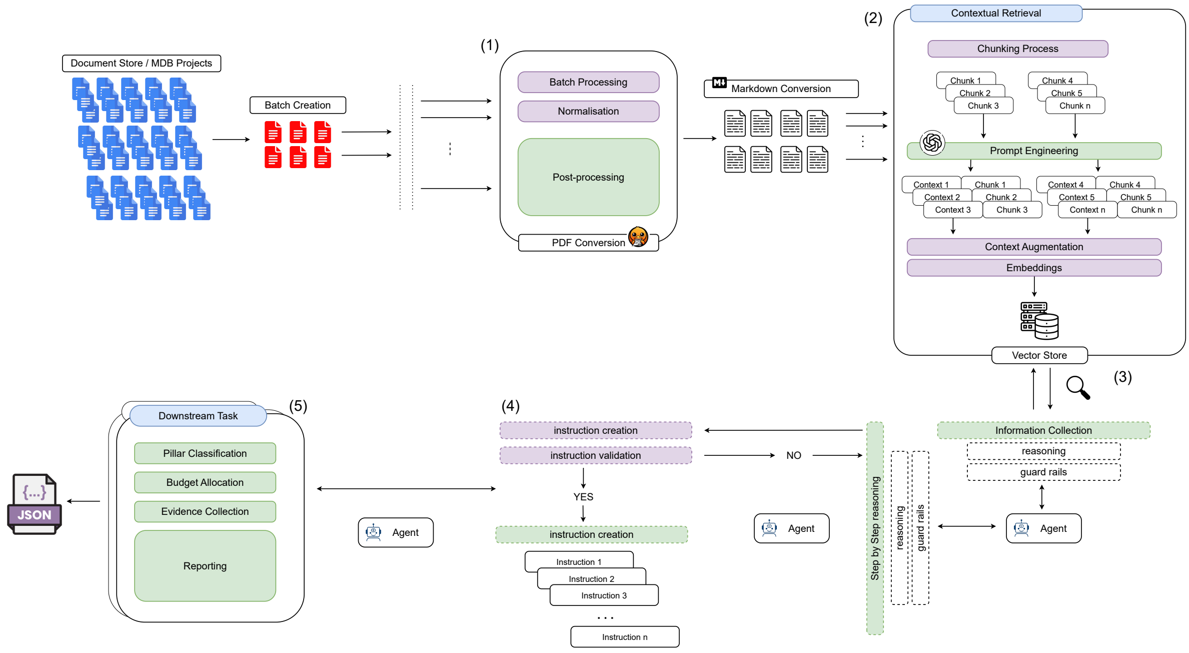


Figure 1: AI-driven financial tracking pipeline for EWS investments. The workflow comprises five stages: (1) PDF conversion using Docling parser, (2) context-augmented chunking with document-level summaries, (3) hybrid retrieval combining dense vectors and BM25F lexical search, (4) iterative agent-based sub-query generation with self-validation loops, and (5) downstream task execution including pillar classification and budget allocation with evidence grounding.

4. **Export Functionality:** Users can export structured JSON outputs for downstream analysis, integration with existing financial tracking systems, or portfolio-level aggregation.

4.2 Key Features

Evidence Traceability. Each budget allocation is linked to specific text or table fragments from the source document. Users can click on any pillar allocation to view the supporting evidence, enabling expert validation and audit trails.

Confidence Indicators. The system provides confidence scores for each classification decision, flagging low-confidence predictions that may require human review. This supports the expert-in-the-loop workflow essential for financial accountability.

Multi-Document Analysis. Users can upload multiple documents for batch processing, enabling portfolio-level analysis across projects or funds. Aggregated views show cross-document patterns and potential gaps in EWS coverage.

Comparison Mode. For research and validation purposes, the interface offers a comparison view showing outputs from different system configurations (e.g., agent-based vs. baseline methods) side-by-side.

4.3 Technical Architecture

The deployed system comprises:

- **Frontend:** React-based web application hosted on Vercel, providing responsive UI and real-time updates via WebSocket connections.
- **Backend:** FastAPI server handling document processing, agent orchestration, and API endpoints.
- **Vector Database:** Weaviate instance for efficient hybrid retrieval over indexed document embeddings.
- **LLM Integration:** OpenAI API for reasoning and classification tasks, with configurable model selection.

The system processes a typical 50-page MDB project document in under 3 minutes, compared to 2–3 hours for manual expert analysis—a reduction of over 98% in processing time.

5 Results

Evaluation Protocol: Unless stated otherwise, we evaluate on held-out test sets split at the document level, so that no project report contributes evidence to more than one split. For the pillar-level experiment, the classifier and baselines are trained and tuned on a subset of CREWS-Fund documents and evaluated on disjoint projects; for the MDB Evidence Set, the evidence segments in the test split are drawn exclusively from held-out documents. This prevents label leakage across splits and ensures that performance is measured on previously unseen reports (see Appendix E for split statistics and sampling details).

5.1 Pillar-Level Budget Classification

We frame the CREWS-Fund experiment as a joint pillar-classification and budget-allocation task. For each document d we observe a gold *pillar budget vector*

$$\mathbf{b}_d = \big(b_{d,1},\dots,b_{d,5}\big) \in \mathbb{R}^5_{\geq 0}, \qquad \sum_{p=1}^{5} b_{d,p} = B_d^{\text{tot}}, \tag{1}$$

where $b_{d,p}$ is the amount assigned to EWS pillar p and B_d^{tot} is the total EWS envelope. Gold budgets satisfy the conservation constraint by construction; model predictions $\hat{b}_{d,p}$ are not renormalized and may over- or under-allocate across pillars.

Binary pillar indicators are defined as

$$y_{d,p} = [\![b_{d,p} > 0]\!] \in \{0,1\}, \tag{2}$$

with Iverson bracket $[\![\cdot]\!]$. The model outputs $\hat{\mathbf{b}}_d$ and $\hat{y}_{d,p} = [\![\hat{b}_{d,p} > 0]\!]$. Aggregation of chunk-level outputs into document-level $\hat{\mathbf{b}}_d$ and $\hat{y}_{d,p}$ is defined in Appendix G.5.

A prediction for pillar p in document d is a true positive (TP) only if

(a) **Label correct:** $y_{d,p} = 1$ and $\hat{y}_{d,p} = 1$;

(b) **Budget within tolerance:**

$$\left|\hat{b}_{d,p} - b_{d,p}\right| \leq 0.05\, B_d^{\text{tot}}, \tag{3}$$

i.e., a ±5% window around the gold pillar amount.

If the model predicts a pillar where $y_{d,p} = 0$ or violates (3), we count a false positive (FP); if $y_{d,p} = 1$ but the pillar is missing or outside the tolerance, we count a false negative (FN). We compute Accuracy, Precision, Recall, and F_1 over all (d,p) pairs and report macro-averaged scores across pillars.

Using a manually annotated CREWS-Fund corpus (Appendix E), we benchmark four baselines (Zero-Shot, Few-Shot, Transformer, Few-Shot-CoT) against our *Glass-Box Agentic* pipeline. As shown in Table 1, the agent attains $\mathbf{0.87}$ accuracy, $\mathbf{0.89}$ precision, and $\mathbf{0.83}$ recall, an 8–14 pp improvement over the strongest baseline.

The evaluation set reflects the imbalanced distribution of pillars and budget magnitudes that analysts encounter in practice, rather than an artificially balanced benchmark.

Method	**Accuracy**	**Precision**	**Recall**
Zero-Shot	0.41	0.40	0.61
Few-Shot	0.42	0.45	0.64
Transformer	0.41	0.64	0.32
Few-Shot-CoT	0.51	0.63	0.71
Agent	**0.87**	**0.89**	**0.83**

Table 1: Evaluation metrics for budget distribution across the EWS Pillars. The agent-based approach significantly outperforms all baselines on all metrics.

These figures show that the agent not only identifies the correct set of pillars but also assigns budget to them with tight numeric fidelity, providing a solid reference line for the broader Glass-Box vs. Black-Box study in § 5.2.

5.2 Glass-Box *vs.* Black-Box Study (MDB Evidence Set)

To assess whether transparency still pays off in an end-to-end setting, we construct an expert-annotated MDB evidence set co-curated with the World Meteorological Organization (WMO) (see Appendix E). Each segment is labeled with its EWS pillar, the corresponding budget amount, the evidence–pillar linkage, and the document's total EWS budget, allowing us to jointly evaluate retrieval, reasoning traceability, and numerical fidelity.

We compare three systems: our **Glass-Box Agent** (Section A.3), **Gemini 2.5 Flash**, and **OpenAI Assistants**, both used as black-box assistants that process the same PDFs with a single, carefully engineered prompt. For Gemini 2.5 Flash and OpenAI Assistants, the prompt specifies the role (EWS financial analyst), task (EWS funding allocation), the EWS taxonomy, stepwise analysis instructions, and a JSON output schema. Full details and examples are provided in Appendix I.

Performance is evaluated along five facets, using the same aggregation and tolerance rules as in Section 5.1 and Appendix G.5:

- **Evidence extraction:** recall, precision, F_1, and Recall@5 for recovering gold evidence segments.
- **Pillar-label assignment:** multi-label Accuracy, Precision, Recall, and F_1 over the five EWS pillars.
- **Amount distribution across pillars:** comparison of $\hat{b}_{d,p}$ to $b_{d,p}$ with the same ±5%

tolerance band as in Eq. (3), yielding macro-averaged Accuracy, Precision, Recall, and F_1 over (d, p) decisions.

- **Evidence-to-label mapping:** correctness of linking retrieved segments to the right pillar, again via TP/FP/FN counts.
- **Total EWS amount prediction:** for each document

$$\hat{B}_d^{\text{tot}} = \sum_{p=1}^{5} \hat{b}_{d,p},$$

and conservation accuracy

$$\text{acc}_{\text{tot}}(d) = 1 - \frac{|\hat{B}_d^{\text{tot}} - B_d^{\text{tot}}|}{B_d^{\text{tot}}},$$

together with absolute and percentage errors.

The main analysis uses the full, naturally imbalanced evidence set; results on a balanced subsample with equal support per pillar are reported in Appendix I, Table 5.

5.3 Interpretation of the Benchmark

Total-amount accuracy (Fig. 2, left). The Glass-Box Agent attains the highest median total-amount accuracy ($\tilde{x} \approx 0.78$) with a narrow interquartile range, indicating stable performance across heterogeneous layouts. Gemini 2.5 Flash and OpenAI Assistants trail behind (median ≈ 0.73 and ≈ 0.68) and exhibit heavier tails, reflecting more frequent large conservation errors.

Amount-per-pillar performance (Fig. 2, right). When accuracy is measured at the pillar level, the Agent captures nearly half of the aggregate macro-F_1 mass (48.7%), while Gemini 2.5 Flash accounts for 36.1% and OpenAI Assistants 15.2%. This mirrors Table 1: schema-aware, transparent reasoning yields the most faithful pillar-level budget breakdowns.

Evidence-extraction robustness (Fig. 3). Across most MDB projects, the Agent attains the highest evidence-extraction F_1, with Gemini 2.5 Flash and OpenAI trailing. The main exception are where budgets are not in explicit tables but diffused through narrative text (grey bands), where Gemini 2.5 Flash slightly outperforms the Agent, reflecting a residual advantage of large black-box models on heavily prose-centric layouts; this is consistent with the balanced-subsample scores in Table 5 (Appendix I), where the Agent still leads overall.

The benchmark indicates that *glass-box, modular retrieval–reasoning pipelines* dominate on structured and semi-structured financial disclosures, while black-box assistants narrow the gap only when numeric cues are deeply embedded in free-form text. Closing this gap is a key direction for future work, for example by enriching the Agent's retrieval module with paragraph-level numerical parsing.

5.4 Ablation Study

To quantify the contribution of individual components of the Glass-Box Agent, we conduct an ablation study on the MDB evidence development set. We systematically remove (i) context augmentation, (ii) hybrid dense+BM25F retrieval, (iii) the top-k setting used for retrieval, and (iv) the agent's self-healing loop, measuring the impact on evidence extraction F_1, Recall@5, pillar-level macro-F_1, and total-amount accuracy.

Removing context augmentation yields a noticeable drop in retrieval quality and downstream budget fidelity, confirming that short document-level summaries help disambiguate otherwise similar chunks. Switching from hybrid to dense-only retrieval primarily hurts Recall@5 and evidence F_1, indicating that exact lexical matching is still crucial for capturing scattered numerical clues. Varying k shows that $k = 5$ provides the best trade-off between coverage and noise. Finally, disabling the self-healing loop (single-pass retrieval with no re-querying) reduces both evidence F_1 and total-amount accuracy, particularly on documents with fragmented tables, underscoring the importance of iterative verification. Full ablation results are reported in Appendix J.

6 Bias Awareness and Mitigation

We acknowledge that our system may exhibit biases inherited from training data, particularly when classifying novel financial structures or terminology not well-represented in the CREWS Fund corpus. To address these concerns, we implement several mitigation strategies:

Confidence-Based Human Review. The system outputs confidence scores for each pillar classification. Predictions with confidence below a configurable threshold (default: 0.7) are automatically flagged for human expert review, ensuring that uncertain classifications do not propagate unchecked.

Pillar-Level Uncertainty Quantification. Beyond point predictions, we compute uncertainty esti-

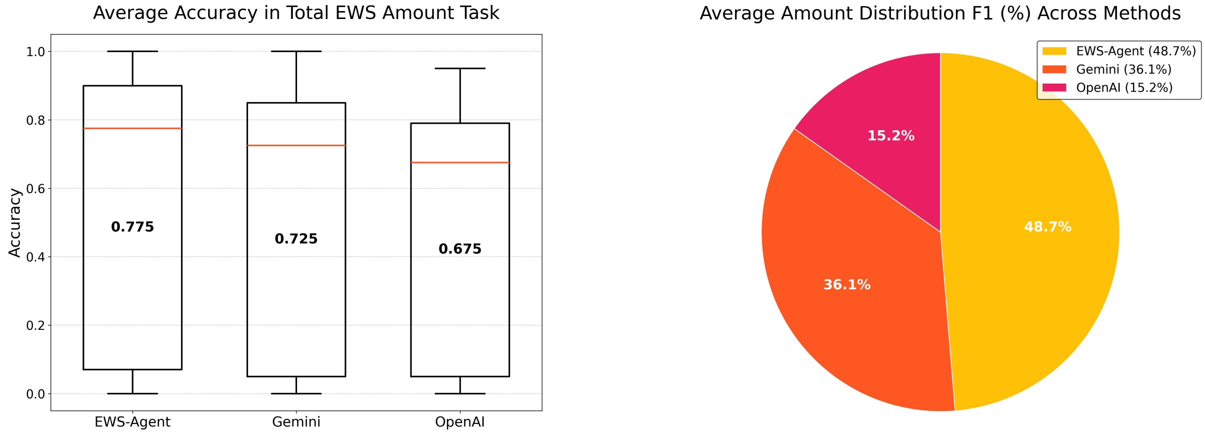


Figure 2: Left: box-plot of the average accuracy on the Total EWS Amount task, evaluated on the expert-annotated test set for each system. The Glass-Box Agent shows the highest median accuracy with the narrowest inter-quartile range, indicating consistent performance. Right: Pie chart showing each system's share of the overall macro-averaged F1 score on the same test set (EWS-Agent 48.7%, Gemini 36.1%, OpenAI 15.2%).

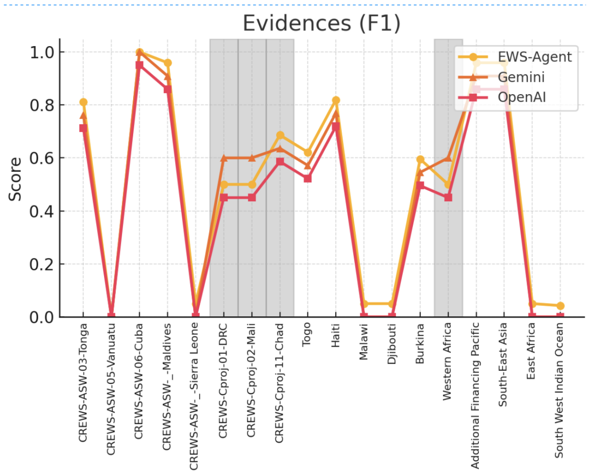


Figure 3: Per-document F_1 for **evidence extraction**. Grey bands highlight projects in which budget figures are dispersed across narrative sections rather than formatted tables. The Glass-Box Agent (yellow) consistently outperforms black-box alternatives except in heavily prose-centric documents.

mates using Monte Carlo dropout during inference. High-uncertainty predictions are highlighted in the user interface, enabling analysts to prioritize review efforts.

Expert-in-the-Loop Validation. The deployed system (Section 4) supports an expert validation workflow where domain specialists can review, validate, and optionally override system predictions. All corrections are logged, creating a feedback loop for continuous model improvement.

Cross-Fund Generalization Testing. While our primary evaluation uses CREWS Fund documents, we conducted preliminary tests on documents from other climate funds (Green Climate Fund, Adaptation Fund) to assess generalization. Performance degradation on out-of-distribution documents is documented in Appendix K, and we recommend re-calibration when applying the system to new funding sources.

Terminology Coverage Analysis. We maintain a glossary of EWS-related terms encountered during training and flag documents containing significant out-of-vocabulary terminology. This alerts users when the system encounters potentially novel financial structures.

7 Conclusion

We presented the EW4All Financial Tracking AI-Assistant, an agent-based RAG system designed to extract EWS investments from heterogeneous MDB reports. Achieving 87% accuracy on a manually annotated corpus, our approach significantly outperforms traditional NLP baselines and provides a transparent alternative to black-box assistants. The system is publicly deployed and currently supports early adopters in uncovering uncatalogued investments and accelerating reporting; we refer readers to Appendix C for full deployment details, real-world impact case studies, and future research directions.

References

Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Nikolaos Livathinos, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Fabian Lindlbauer, Kasper Dinkla, Lokesh Mishra, Yusik Kim, Shubham Gupta, Rafael Teixeira de Lima, Valery Weber, Lucas Morin, Ingmar Meijer, Viktor Kuropiatnyk, and Peter W. J. Staar. 2024. Docling technical report.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.

Gordon V. Cormack, Charles L.A. Clarke, and Stephan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 758–759. ACM.

Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. 2024. Active prompting with chain-of-thought for large language models.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges.

Michael Günther, Isabelle Mohr, Daniel James Williams, Bo Wang, and Han Xiao. 2024. Late chunking: Contextual chunk embeddings using long-context embedding models.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Junwei Liu, Kaixin Wang, Yixuan Chen, Xin Peng, Zhenpeng Chen, Lingming Zhang, and Yiling Lou. 2024. Large language model-based agents for software engineering: A survey.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic embed: Training a reproducible long context text embedder.

Gianluca Pescaroli, Sarah Dryhurst, and Georgios Marios Karagiannis. 2025. Bridging gaps in research and practice for early warning systems: new datasets for public response. *Frontiers in Communication*, 10:1451800.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Andrew C Tupper and Carina J Fearnley. 2023. Mind the gaps in disaster early-warning systems—and fix them. *Nature*, 623:479.

Omar Velazquez, Gianluca Pescaroli, Gemma Cremen, and Carmine Galasso. 2020. A review of the technical and socio-organizational components of earthquake early warning systems. *Frontiers in Earth Science*, 8:533498.

Jie Wang, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M Jose. 2024. Reinforcement learning-based recommender systems with large language models for state reward and action modeling. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 375–385.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. 2023. The rise and potential of large language model based agents: A survey.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

A Detailed System Methodology

This appendix provides the technical details of the implementation used in the main paper's System Overview.

A.1 Embedding Construction and Indexing

Effective downstream reasoning over MDB PDFs requires an embedding index that respects heterogeneous layouts and scattered evidence. We therefore use a five-stage pipeline: document parsing, chunking, context augmentation, embedding generation, and vector storage.

First, we extract raw text and structural elements from each document d with the Docling converter (Auer et al., 2024):

$$T_d = \text{DoclingParser}(d), \tag{4}$$

where T_d denotes all extracted textual elements (narrative segments, tables, and other layout blocks). We then partition T_d into disjoint chunk sets

$$\mathcal{C} = \mathcal{C}_{\text{struct}} \cup \mathcal{C}_{\text{text}}, \tag{5}$$

where $\mathcal{C}_{\text{struct}}$ contains tables and structured components (e.g., headers, multi-column regions) and $\mathcal{C}_{\text{text}}$ contains narrative passages and other non-tabular blocks. This separation preserves structural boundaries and avoids flattening tables or merging unrelated segments, which would degrade embedding quality and retrieval.

To situate each chunk in its document context and reduce semantic ambiguity (Günther et al., 2024), we generate a short summary for each $c \in \mathcal{C}$ by prompting an LLM with $P_{\text{ctx}}(c, T_d)$:

$$\text{ctx}(c) \;=\; \text{LLM}\big(P_{\text{ctx}}(c, T_d)\big), \tag{6}$$

and form the augmented chunk

$$c' = c \;\oplus\; \text{ctx}(c). \tag{7}$$

All augmented chunks c' are encoded in a single latent space:

$$e_{\text{tt}}(c') = f_{\text{tt}}(c') \in \mathbb{R}^{d_{\text{tt}}}, \tag{8}$$

where f_{tt} is a joint text–structure encoder. We empirically compared `bge-m3` (Chen et al., 2024), `nomic-embed-text:v1.5` (Nussbaum et al., 2024), and OpenAI's `text-embedding-3-small`, and selected `text-embedding-3-small` based on Recall@5, nDCG@5, and MRR@5 on the MDB evidence set (Table 4, Appendix F).

Embeddings are indexed in a Weaviate environment with separate NamedVector configurations for text and structured layouts, enabling efficient hybrid (semantic + lexical) search. Each embedding $e(c')$ is stored with metadata:

$$\text{VDB\_store}\big(e(c'),\, \text{meta}(c')\big). \tag{9}$$

At inference time, for a file ID f and query q, we retrieve the top-5 relevant chunks:

$$\mathcal{R}(f) \;=\; \text{VDB\_query}(q,\, f), \quad |\mathcal{R}(f)| = 5. \tag{10}$$

Further details on embedding model selection, Weaviate configuration, and chunk metadata are provided in Appendix F.

A.2 Hybrid Retrieval via Rank Fusion

In addition to the above procedure, we employ a hybrid search strategy that combines dense vector search with BM25F-based keyword search (Robertson and Zaragoza, 2009) to leverage both semantic similarity and exact lexical matching. Let $\mathcal{R}_v(q, f)$ denote the set of candidate chunks retrieved via dense vector search, and let $\mathcal{R}_k(q, f)$ denote the candidate chunks obtained via BM25F keyword search. To fuse these two retrieval sets, we use Reciprocal Rank Fusion (RRF) (Cormack et al., 2009). For each candidate chunk $c \in \mathcal{R}_v(q, f) \cup \mathcal{R}_k(q, f)$, we compute an RRF score as:

$$\text{RRF}(c) = \sum_{i \in \{v,k\}} \frac{1}{\text{rank}_i(c) + K}, \tag{11}$$

where $\text{rank}_i(c)$ is the rank of c in retrieval system i (with lower ranks corresponding to higher relevance) and K is a smoothing constant (typically set to 60).

In cases where candidates from different retrieval systems share the same RRF score (e.g., when top-3 candidates from each method have no overlap and their 3rd-ranked chunks yield identical scores), we apply a secondary sort by dense vector similarity score to break ties deterministically.

The final set of retrieved chunks is then given by selecting the top five candidates according to their RRF scores:

$$\mathcal{R}(f) = \text{Top5}\Big(\mathcal{R}_v(q, f) \cup \mathcal{R}_k(q, f),\, \text{RRF}(c)\Big). \tag{12}$$

This hybrid method harnesses the semantic sensitivity of dense vector retrieval alongside the precise lexical matching of BM25F.

A.3 Classification and Budget Allocation

For each retrieved chunk $c' \in \mathcal{R}(f)$, we predict an EWS pillar label vector y (over the five pillars) and an associated budget B. We compare four baselines that differ in how they obtain y and B, plus our agent-based approach; implementation details are provided in Appendix G.

Zero-Shot and Few-Shot Classification

In the zero-shot and few-shot baselines, we construct a prompt $P_{\text{Class+Budget}}(c')$ that includes the augmented chunk (and, in the few-shot case, a small set of labeled examples). The LLM directly outputs both labels and budget:

$$\{y, B\} = \text{LLM}(P_{\text{Class+Budget}}(c')). \quad (13)$$

This method leverages the pre-trained knowledge of the LLM, with few-shot prompting guiding its responses.

Fine-Tuned Transformer-Based Classifier

As a classical NLP baseline, we fine-tune a BERT-base encoder M_{ft} as a multi-label classifier on the labeled chunks $\{(c'_i, y_i)\}_{i=1}^{N}$ (see Appendix G.2 for details on the architecture). The model outputs a 5-dimensional sigmoid layer and yields pillar predictions $y = M_{\text{ft}}(c')$. Budgets are then inferred by a separate LLM call:

$$B = \text{LLM}(P_{\text{Budget}}(c', y)). \quad (14)$$

Chunk-level $\{y, B\}$ tuples are later aggregated to document-level budgets as described in Appendix H, and conservation is evaluated only at aggregation time via the document-level metrics in Section 5.

Few-Shot CoT Classification

This approach employs a three-step Chain-of-Thought (CoT) strategy, resulting in a tuple $\{y, B\}$. First, structured-layout (e.g., tables) chunks are optionally reformatted into clean markdown: $c'' = \text{LLM}(P_{\text{reformat}}(c'))$, otherwise, we set $c'' = c'$. Second, we classify the (reformatted) chunk: $y = \text{LLM}(P_{\text{Class}}(c''))$. Third, we allocate the budget conditioned on both content and labels: $B = \text{LLM}(P_{\text{Budget}}(c'', y))$. This CoT-style factorization encourages more explicit reasoning over table structure and pillar definitions; full prompts and examples are in Appendix G.3.

Agent-Based Approach

Our agent-based method replaces fixed prompts with an LLM agent that plans, retrieves, and validates before emitting $\{y, B\}$. Given a document f, the agent executes the following steps:

1. **Planning:** Generate a set of sub-tasks $I = \{i_1, \dots, i_k\}$ and retrieval queries $Q = \{q_1, \dots, q_\ell\}$.
2. **Retrieval:** Issue vector-database queries $\text{VDB\_query}(q, f)$ for each relevant sub-task.
3. **Self-validation:** Check coverage sufficiency; re-query when thresholds are unmet:

$$c'_{i_j}{}^{\text{final}} = \begin{cases} \text{VDB\_query}(q_{i_j}^{\text{new}}, f), & \text{if } c'_{i_j} \text{ insufficient,} \\ c'_{i_j}, & \text{otherwise.} \end{cases} \quad (15)$$

4. **Consolidation:** Aggregate intermediate results into a schema-aligned JSON output $\{y, B\}$ per chunk and document.

The full agent loop, instruction format, and guardrails are detailed in Appendix G.4.

B System Demonstration Details

B.1 Accessing the System

The EW4All Financial Tracking AI-Assistant is publicly accessible at:

https://ews-front.vercel.app/

The system requires no installation and runs entirely in the browser. Users can create accounts to save analysis history and export results.

B.2 System Requirements

- Modern web browser (Chrome, Firefox, Safari, Edge)
- PDF documents up to 100 pages
- Stable internet connection for API calls

B.3 API Access

For programmatic access and integration with existing workflows, we provide a REST API. Documentation is available at https://ews-front.vercel.app/api/docs. The API supports:

- Document upload and processing
- Batch analysis of multiple documents
- Retrieval of structured JSON outputs
- Webhook notifications for async processing

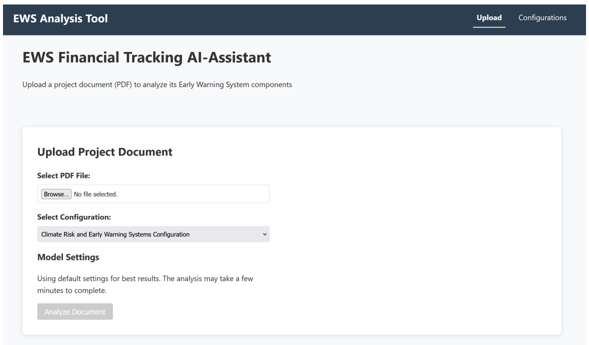
B.4 Sample Outputs

Figure 4 shows a sample analysis report generated by the system, illustrating the structured output format with pillar allocations, evidence links, and confidence scores.

C Real-World Deployment and Impact

Since our agent-based RAG pipeline was deployed in March 2024, early adopters in the EWS community have realized significant benefits:

- **Uncovering hidden investments.** The World Meteorological Organization (WMO) used the system to scan its MDB portfolio, identifying dozens of EWS allocations that had not



EWS Analysis Tool

EWS Financial Tracking AI-Assistant

Upload a project document (PDF) to analyze its Early Warning System components

Upload Project Document

Select PDF File: No file selected.

Select Configuration: Climate Risk and Early Warning Systems Configuration

Model Settings

Using default settings for best results. The analysis may take a few minutes to complete.

Figure 4: Sample analysis report from the deployed system showing pillar-wise budget allocation with evidence grounding.

previously been catalogued in their internal records.

- **Driving reporting guideline enhancements.** Drawing on classification gaps revealed by our model, the CREWS Fund updated its grant-reporting templates to standardize pillar-level expenditure tagging.
- **Accelerating analysis throughput.** Automated processing reduced the time per project report from 2–3 hours manually to under 3 minutes end-to-end, freeing analysts to focus on higher-value tasks.

These case studies illustrate how transparent, automated extraction not only boosts operational efficiency but also informs better policy and accountability practices at multilateral development banks and climate funds.

D Early Warning Systems (EWS)

D.1 Definition and Purpose

Early Warning Systems (EWS) are integrated frameworks designed to detect imminent hazards and alert authorities and communities before disasters strike. In essence, an EWS combines hazard monitoring, risk analysis, communication, and preparedness planning to enable timely, preventive actions. Early warnings are a cornerstone of disaster risk reduction (DRR) – they save lives and reduce economic losses by giving people time to evacuate, protect assets, and secure critical infrastructure[2]. By empowering those at risk to act ahead of a hazard, EWS help build climate resilience: they are proven to safeguard lives, livelihoods, and ecosystems amid increasing climate-related threats[3]. In summary, an effective EWS ensures that impending dangers are rapidly identified, warnings reach the impacted population, and appropriate protective measures are taken in advance.

D.2 EWS Taxonomy

A robust EWS involves several fundamental components that work together seamlessly. The United Nations identify four interrelated pillars necessary for an effective people-centered EWS (Pescaroli et al., 2025). This taxonomy serves as a structured framework to categorize EWS components and activities, facilitating a consistent approach to analyzing early warning systems across various domains. Our approach in this paper is based on these four fundamental pillars of EWS and one cross-pillar, ensuring a comprehensive understanding of risk knowledge, detection, communication, and preparedness.

Early Warning System (EWS) Taxonomy Prompt

An Early Warning System (EWS) is an integrated system of hazard monitoring, forecasting, and prediction, disaster risk assessment, communication, and preparedness activities that enables individuals, communities, governments, businesses, and others to take timely action to reduce disaster risks before hazardous events occur.
When analyzing a text, it is essential to determine whether it falls under EWS components and activities, which vary across multiple sectors and require coordination and financing from various actors.
The taxonomy is based on the Four Pillars of Early Warning Systems and one cross-pillar:

Pillar 1: Disaster Risk Knowledge and Management (Led by UNDRR)

This pillar focuses on understanding disaster risks and enhancing the knowledge of communities by collecting and utilizing comprehensive information on hazards, exposure, vulnerability, and capacity.

[2] See https://www.unisdr.org/files/608_10340.pdf.

[3] See https://www.unep.org/topics/climate-action/climate-transparency/climate-information-and-early-warning-systems.

Illustrative examples:

- Inclusive risk knowledge: Incorporating local, traditional, and scientific risk knowledge.
- Production of risk knowledge: Establishing a systematic recording of disaster loss data.
- Risk-informed planning: Ensuring decision-makers can access and use updated risk information.
- Data rescue: Digitizing and preserving historical disaster data.

Keywords: Risk mapping, vulnerability mapping, disaster risk reduction (DRR), climate information.

Pillar 2: Detection, Observation, Monitoring, Analysis, and Forecasting (Led by WMO)

This pillar enhances the capability to detect and monitor hazards, providing timely and accurate forecasting.

Illustrative examples:

- Observing networks enhancement: Strengthening real-time monitoring systems.
- Hazard-specific observations: Improving monitoring of high-impact hazards.
- Impact-based forecasting: Developing quantitative triggers for anticipatory action.

Keywords: Forecasting, seasonal predictions, multi-model projections, climate services.

Pillar 3: Warning Dissemination and Communication (Led by ITU)

Effective communication ensures that early warnings are received by those at risk, enabling them to take timely action.

Illustrative examples:

- Multichannel alert systems: Use of SMS, satellite, sirens, and social media.
- Standardized warnings: Implementation of the Common Alerting Protocol (CAP).
- Feedback mechanisms: Enabling community input on warning effectiveness.

Keywords: Communication systems, multichannel dissemination, emergency broadcast systems.

Pillar 4: Preparedness and Response Capabilities (Led by IFRC)

Timely preparedness and response measures translate early warnings into life-saving actions.

Illustrative examples:

- Emergency preparedness planning: Developing anticipatory action frameworks.
- Public awareness campaigns: Educating communities on disaster response.
- Emergency shelters: Construction of cyclone shelters, evacuation centers.

Keywords: Preparedness planning, emergency drills, public education on disaster response.

Cross-Pillar: Foundational Elements for Effective EWS

Cross-cutting elements critical to the sustainability and effectiveness of EWS include governance, inclusion, institutional arrangements, and financial planning.

Illustrative examples:

- Governance and institutional frameworks: Defining roles of agencies and stakeholders.
- Financial sustainability: Mobilizing and tracking finance for early warning systems.
- Regulatory support: Developing and enforcing data-sharing legislation.

Keywords: Institutional frameworks, governance, financial sustainability, data management.

Each of these components is vital. Only when risk knowledge, monitoring, communication, and preparedness work in unison can an early warning system effectively protect lives and properties. Gaps in any one element (for example, if warnings don’t reach the vulnerable, or if communities don’t know how to respond) will weaken the whole system. Thus, successful EWS are people-centered and end-to-end, linking high-tech hazard detection with on-the-ground community action.

D.3 Importance for Climate Finance

EWS are widely recognized as a high-impact, cost-effective investment for climate resilience. By providing advance notice of floods, storms, heatwaves and other climate-related hazards, EWS significantly reduce disaster losses. Studies indicate that every $1 spent on early warnings can save up to $10 by preventing damages and losses.[4] For example, just 24 hours' warning of an extreme event can cut ensuing damage by about 30%, and an estimated USD $800 million investment in early warning infrastructure in developing countries could avert $3–16 billion in losses every year[5]. These economic benefits underscore why EWS are considered "no-regret" adaptation measures, i.e., they pay for themselves many times over by protecting lives, assets, and development gains.

Given their proven value, EWS have become a priority in climate change adaptation and disaster risk reduction funding. International climate finance mechanisms, such as the Green Climate Fund, Climate Risk and Early Warning Systems (CREWS) Fund, and Adaptation Fund along with development banks, are channeling resources into EWS projects, from modernizing meteorological services and hazard monitoring networks to community training and alert communication systems. Strengthening EWS is also central to global initiatives like the United Nations' Early Warnings for All (EW4All), which calls for expanding early warning coverage to 100% of the global population by 2027. Achieving this goal requires substantial financial support to build new warning systems in climate-vulnerable countries and to maintain and upgrade existing ones. Climate finance is therefore being directed to help develop, implement, and sustain EWS, ensuring that countries can operate these systems (e.g., funding for equipment, data systems, and personnel) over the long term.

In summary, investing in EWS is essential for climate resilience. It not only reduces humanitarian and economic impacts from extreme weather, but also yields high returns on investment. Financial support for EWS, whether through dedicated climate funds, loans and grants, or public budgets, underpins their development and sustainability, making it possible to deploy cutting-edge technology and foster prepared communities. By mitigating the worst effects of climate disasters, EWS help safeguard development progress, which is why they feature prominently in climate adaptation financing and strategies.

D.4 Current Challenges

Despite their clear benefits, there are several challenges in financing and implementing EWS effectively. Key issues include:

Data Inconsistencies and Lack of Standardization: EWS rely on data from multiple sources (weather observations, risk databases, etc.), but often this data is inconsistent, incomplete, or not shared effectively across systems. Differences in how hazards are monitored and reported can lead to gaps or delays in warnings. Likewise, there is a lack of standardization in early warning protocols and data formats between agencies and countries (Velazquez et al., 2020; Pescaroli et al., 2025). Incompatible data systems and inconsistent methodologies (for example, different trigger criteria for warnings or varying risk assessment methods) make it difficult to integrate information. This fragmentation hinders the creation of a "common operating picture" of risk. Data harmonization and common standards (for data collection, forecasting models, and warning communication) are needed to ensure EWS components work together seamlessly.

Institutional and Cross-Organizational Barriers: An effective EWS cuts across many organizations: national meteorological services, disaster management agencies, local governments, international partners, and communities. Coordinating these actors remains a challenge. In many cases, efforts are siloed: meteorological offices may issue technical warnings that don't fully reach or engage local authorities or the public. There are gaps in governance, clarity of roles, and inter-agency communication that can weaken the warning chain. Improving EWS often requires overcoming bureaucratic boundaries and fostering cooperation between different sectors (e.g., linking climate scientists with emergency planners). Interoperability issues—i.e., ensuring different organizations' technologies and procedures align—are also a hurdle (Tupper and Fearnley, 2023). As the World Meteorological Organization (WMO) states, connecting all relevant actors (from international agencies down to community groups) and adapting plans to

[4] See https://wmo.int/news/media-centre/early-warnings-all-advances-new-challenges-emerge.

[5] See https://www.unep.org/topics/climate-action/climate-transparency/climate-information-and-early-warning-systems.

real-world local conditions is complex[6]. Sustained commitment, clear protocols, and partnerships are required to break down these barriers so that EWS operate as a cohesive, cross-sector system.

Financing Gaps and Sustainability: While funding for EWS is rising, it still lags behind what is needed for global coverage and maintenance. Many high-risk developing countries lack the resources to install or upgrade EWS infrastructure (radar, sensors, communication tools) and to train personnel. Fragmented financing is a problem. Support comes from various donors and programs without a unified strategy, leading to potential overlaps in some areas and stark gaps in others. For instance, recent analyses show that a large share of EWS funding is concentrated in a few countries, while Small Island Developing States (SIDS) and Least Developed Countries (LDCs) remain underfunded despite being highly vulnerable[7]. Even when initial capital is provided to set up an EWS, securing long-term funding for operations and maintenance (software updates, staffing, equipment calibration) is difficult. Without sustainable financing, systems can degrade over time. Ensuring financial sustainability, co-financing arrangements, and political commitment is critical so that EWS are not one-off projects but enduring services.

In addition to the above, there are challenges in technological adoption and last-mile delivery: for example, reaching remote or marginalized populations with warnings (issues of language, literacy, and reliable communication channels) and building trust so that people heed warnings. Climate change is also introducing new complexities—hazards are becoming more unpredictable or intense, testing the limits of existing early warning capabilities. Overall, addressing data and standardization issues, improving institutional coordination, and closing funding gaps are priority challenges to fully realize the life-saving potential of EWS.

D.5 Relevance to This Study

Our work is focused on the financial tracking and classification of investments in climate resilience, and EWS represent a prime example of such investments. Early warning projects often cut across sectors and funding sources—they might include components of infrastructure, technology, capacity building, and community outreach. Because of this cross-cutting nature, tracking where and how money is spent on EWS can be difficult without a clear classification system. Different organizations may label EWS-related activities in various ways (e.g., "hydromet modernization", "disaster preparedness", "climate services"), leading to inconsistencies in investment data. By establishing a standardized framework to define and categorize EWS investments, the study helps create a "big-picture view" of early warning financing. This enables analysts and policymakers to identify overlaps, gaps, and trends that were previously obscured by fragmented data.

Moreover, improving the classification of EWS funding directly supports broader resilience initiatives. For instance, the newly launched Global Observatory for Early Warning System Investments is already working to tag and track EWS-related expenditures across major financial institutions. Such efforts mirror the goals of this study by highlighting the need for consistent tracking, transparency, and coordination in climate resilience finance. Better classification of investments means stakeholders can pinpoint where resources are going and where additional support is needed to meet global targets like the "Early Warnings for All by 2027" pledge. In short, EWS feature in this study as a critical category of climate resilience investment that must be clearly identified and monitored.

By including EWS in its financial tracking framework, the study provides valuable insights for decision-makers. It helps determine how much funding is allocated to early warnings, from which sources, and for what components (equipment, training, maintenance, etc.). This information is crucial for evidence-based decisions on scaling up EWS: for example, spotting a shortfall in community-level preparedness funding, or recognizing successful investment patterns that could be replicated. Ultimately, linking EWS to the study's financial tracking reinforces the message that climate resilience investments can be better managed when we know their size, scope, and impact area. By classifying EWS expenditures systematically, the study contributes to stronger accountability and strategic planning in building climate resilience, ensuring that early warning systems—and the communities they protect—get the support they urgently need.

[6] See https://wmo.int/news/media-centre/early-warnings-all-advances-new-challenges-emerge.

[7] See https://wmo.int/media/news/tracking-funding-life-saving-early-warning-systems.

E Dataset Construction

In this study, we analyze financial information extracted from MDB project PDFs that contain both structured and unstructured data. Unlike conventional benchmark datasets, these documents exhibit high heterogeneity in their formats: some tables are well-structured, while others embed financial figures within free-text paragraphs or disperse them across multiple rows and columns. In many cases, a single numerical value corresponds to several rows or sub-rows within the same column, creating challenges for extraction, alignment, and interpretation.

E.1 CREWS-Fund Budget Corpus (Pillar-Level)

The pillar-level budget experiment is based on a corpus of 500 CREWS-Fund project reports, with a total of 20,000 expert-annotated segments. We split this corpus at the *document* level into training, validation, and test sets with a 70/20/10 proportion; no project appears in more than one split, preventing cross-document leakage. The label distribution is intentionally imbalanced and mirrors real-world practice: some EWS pillars receive substantially more annotated budget than others, and many projects assign zero budget to certain pillars.

The annotated data, provided by domain experts in CSV format, together with the corresponding PDFs, are included in the supplementary materials of this paper. Each row in the CSV file contains the following nine fields: *Fund*, *Project ID*, *Component*, *Outcome/Expected-Outcome/Objectives*, *Output/Sub-component*, *Activity/Output Indicator*, *Page Number*, *Amount*, and *Label*. The total amount of Early Warning Systems (EWS) funding for a given project is computed as the sum of all *Amount* values associated with that project.

E.2 Dataset Statistics

Table 2 summarizes the key statistics of our annotated corpus.

Statistic	Value
Total documents	500
Total annotated segments	20,000
Training set (documents)	350 (70%)
Validation set (documents)	100 (20%)
Test set (documents)	50 (10%)
Average segments per document	40
Average pages per document	47

Table 2: Dataset statistics for the CREWS-Fund corpus.

E.3 Pillar Distribution

The distribution of annotations across EWS pillars reflects real-world funding patterns:

Pillar	Segments	Percentage
Pillar 1 (Risk Knowledge)	3,200	16%
Pillar 2 (Detection/Forecasting)	6,400	32%
Pillar 3 (Dissemination)	4,000	20%
Pillar 4 (Preparedness)	4,800	24%
Cross-Pillar (Governance)	1,600	8%

Table 3: Distribution of annotated segments across EWS pillars.

E.4 Data Access and Licensing

The annotated corpus (CSV file and PDFs) consists of financial reports and investment documents sourced from publicly available institutional records, which are intended for public information, research, and transparency purposes. The dataset is used strictly within this intended scope—analyzing financial tracking in climate investments—and adheres to the original access conditions. For all artifacts derived from this corpus, including benchmark datasets and classification models, we explicitly specify their intended use for research and evaluation in automated financial tracking and ensure compliance with relevant ethical research guidelines.

F Embedding Model Selection

To select the joint text–table encoder f_{tt}, we constructed a small retrieval benchmark from MDB project documents. For each annotated evidence segment, we issued the corresponding query and measured retrieval quality on a held-out development split. We report standard top-k metrics: Recall@5, nDCG@5, and MRR@5, computed over all queries.

Table 4 summarizes the results for the three candidate encoders. OpenAI's `text-embedding-3-small` achieves the best performance across all metrics, and we therefore use it as f_{tt} in all experiments.

F.1 Weaviate Configuration

We deploy a Weaviate cluster with:

- Two NamedVectors per object: one for $e_{\mathrm{tt}}(c')$ (semantic) and one for a bag-of-words representation (lexical).
- HNSW indexing for the semantic vector, with tuned `efConstruction` and `M` parameters.

Encoder	R@5	nDCG@5	MRR@5
`bge-m3`	0.72	0.68	0.65
`nomic-embed`	0.70	0.66	0.63
OpenAI `text-embedding-3-small`	**0.78**	**0.73**	**0.70**

Table 4: Retrieval performance of candidate embedding models on the MDB evidence development set. OpenAI's `text-embedding-3-small` achieves the best overall ranking quality and is used in our deployed system.

- BM25 configuration for lexical search, used in parallel with vector retrieval.

Hybrid scores are formed by a weighted combination of semantic and lexical similarity; weights were chosen on a small dev set to maximize Recall@5.

F.2 Chunk Metadata

The metadata $\mathrm{meta}(c')$ stored with each embedding (Eq. 9) includes:

- Document identifier f and page number,
- Chunk type (structured vs. text) and original layout coordinates,
- Section title and table caption (when available).

These fields are used for filtering (e.g., table-only retrieval) and for reconstructing human-readable evidence views in the UI.

G Extended Methods: Classification and Budget Allocation

This appendix expands Section A.3, providing full details for each baseline (Zero-Shot / Few-Shot, Fine-Tuned Transformer + LLM, Few-Shot CoT) and for the agent-based system, including prompts, architectures, and training choices that were omitted from the main text for brevity.

G.1 Zero-Shot and Few-Shot Baselines

Prompt structure. For each retrieved chunk $c' \in \mathcal{R}(f)$, we construct a prompt $P_{\text{Class+Budget}}(c')$ with three components:

1. A short description of the task and desired JSON output format for $\{y, B\}$, where y is a 5-dimensional multi-label pillar vector and B is a numeric budget allocation (possibly zero).
2. A concise description of the five EWS pillars, summarized from Appendix D, including 1–2 example activities per pillar.
3. The augmented chunk c' (text or table fragment), enriched with basic metadata (document ID, section title, and page number when available).

Zero-shot variant. In the zero-shot setting, the prompt contains *no* labeled examples: the model relies solely on the pillar descriptions and output schema. The LLM is asked to directly output:

$$\{y, B\} = \mathrm{LLM}(P_{\text{Class+Budget}}(c')), \qquad (16)$$

where $y \in \{0,1\}^5$ (one bit per pillar) and $B \in \mathbb{R}_{\geq 0}$. We enforce the JSON structure with a system-level constraint and discard malformed generations (re-prompting once with an additional format hint).

Few-shot variant. The few-shot variant extends the zero-shot prompt with a small set of K labeled examples $\{(c^{(k)}, y^{(k)}, B^{(k)})\}_{k=1}^{K}$, inserted before the test chunk. Each example includes:

- A short snippet (text or table row/segment) containing a clear EWS signal,
- The gold multi-label vector $y^{(k)}$,
- A corresponding budget value $B^{(k)}$ (or 0 if the snippet does not carry a numeric allocation).

We use $K \in \{3, 5\}$ depending on context length; the examples are chosen to cover all five pillars and a mix of single- and multi-label cases. The few-shot prompt still calls a *single* LLM completion:

$$\{y, B\} = \mathrm{LLM}(P_{\text{Class+Budget}}(c')). \qquad (17)$$

Post-processing. We parse the JSON, map textual pillar names back to indices (`"P1"`–`"P5"`), and clip negative budgets to zero. If the model returns a range (e.g., "USD 0.2–0.3M"), we take the midpoint and convert to a single numeric value in the corpus currency (USD) using the same conversion rules as the annotations.

G.2 Fine-Tuned Transformer + LLM Budget

Model architecture. We fine-tune a BERT-base encoder M_{ft} on labeled chunks $\{(c'_i, y_i)\}_{i=1}^{N}$, where $y_i \in \{0,1\}^5$:

- 12 Transformer layers, hidden size 768, 12 self-attention heads,
- WordPiece/BPE tokenizer with a 30k–50k subword vocabulary,
- Input sequences truncated or padded to 512 subword tokens,
- A 5-dimensional sigmoid output layer:

$$\hat{y} = \sigma(W h_{[\text{CLS}]} + b),$$

where $h_{[\text{CLS}]}$ is the final-layer representation of the `[CLS]` token.

Training objective. We treat pillar prediction as multi-label classification with a class-weighted binary cross-entropy loss:

$$\mathcal{L} = -\sum_{j=1}^{5} w_j \left(y_j \log \hat{y}_j + (1 - y_j) \log(1 - \hat{y}_j) \right), \tag{18}$$

where weights w_j are inversely proportional to pillar frequency in the training split to mitigate label imbalance.

Optimization. We use AdamW with linear learning-rate warmup and decay. A small grid search over learning rate ($\{1\text{e}{-5}, 2\text{e}{-5}, 3\text{e}{-5}\}$) and batch size ($\{8, 16\}$) is performed, selecting the configuration with the best macro-F$_1$ on the development set. Training runs for up to 10 epochs with early stopping based on dev macro-F$_1$.

Thresholding and calibration. We select a global sigmoid threshold τ by maximizing macro-F$_1$ on the dev set, then apply it to obtain binary labels:

$$y_j = \mathbb{I}[\hat{y}_j \geq \tau].$$

Budget allocation prompt. Given the predicted pillar vector y and original chunk c', we invoke a separate LLM call with prompt $P_{\text{Budget}}(c', y)$. The prompt:

1. Reminds the model of the five pillars and provides the predicted subset (e.g., "This chunk is tagged as pillars 2 and 4"),
2. Asks the model to extract the budget amount associated with the EWS-relevant parts of c' (if any),
3. Requests a single numeric value in USD and a short textual justification.

The LLM returns a JSON payload with the numeric budget and explanation; we retain only the numeric field for evaluation.

Aggregation. Chunk-level tuples $\{y, B\}$ are aggregated into document-level labels and budgets following the rules in Appendix G.5.

G.3 Few-Shot CoT Baseline

Reformatting step. To reduce noise from irregular table layouts, we optionally reformat table-like chunks using a prompt $P_{\text{reformat}}(c')$:

$$c'' = \text{LLM}(P_{\text{reformat}}(c')). \tag{19}$$

The prompt asks the model to preserve all numeric entries and column headers, outputting a clean markdown table. For non-table chunks, we set $c'' = c'$.

Pillar classification. We then classify the (possibly reformatted) chunk with a dedicated classification prompt:

$$y = \text{LLM}(P_{\text{Class}}(c'')), \tag{20}$$

which:

- Re-states the five pillar definitions more explicitly than in the zero-shot/few-shot baseline,
- Contains a small number of in-context examples where the model first explains which parts of the text support each pillar and then outputs the final label vector.

The model is instructed to think step by step but only return the final JSON in the answer.

Budget allocation. Finally, we allocate a budget conditioned on both content and labels:

$$B = \text{LLM}(P_{\text{Budget}}(c'', y)). \tag{21}$$

Compared to the simple zero-shot/few-shot baseline, the CoT prompt explicitly asks the model to reason about which lines or cells in the chunk correspond to EWS-related funding, and then to aggregate them into a single amount. The output again consists of a numeric field and a short natural-language rationale.

G.4 Agent-Based System

Instruction schema. The agent operates over a set of high-level instructions $I = \{i_1, \ldots, i_k\}$, where each instruction has:

- `type` (e.g., `FIND_PILLARS`, `EXTRACT_BUDGETS`, `CHECK_CONSERVATION`),
- `inputs` (references to document f, chunk IDs, pillar IDs),
- `outputs` (e.g., list of evidence spans, numeric amounts).

The agent is primed with examples of instruction lists for small documents to illustrate the desired planning behavior.

Planning and query generation. Given a document f, the agent first generates a compact plan:

$$I, Q = \text{LLM}(P_{\text{Plan}}(f\text{\_metadata})),$$

where $Q = \{q_1, \ldots, q_\ell\}$ is a set of retrieval queries. Each instruction i_j may be associated with a specific query q_{i_j} (e.g., "find all chunks related to pillar 2 budgets").

Retrieval and self-validation. For instructions requiring external evidence, the agent issues vector database calls:

$$c'_{i_j} = \text{VDB\_query}(q_{i_j}, f). \tag{22}$$

We then apply a self-validation step where the agent inspects the retrieved chunks and decides whether coverage is sufficient:

$$c'^{\text{ final}}_{i_j} = \begin{cases} \text{VDB\_query}(q^{\text{new}}_{i_j}, f), & \text{if } c'_{i_j} \text{ insufficient,} \\ c'_{i_j}, & \text{otherwise.} \end{cases} \tag{23}$$

Coverage criteria are expressed in natural language in the prompt (e.g., "at least one budget line per pillar mentioned in the document").

Intermediate results. For each instruction i_j, the agent produces an intermediate result result_{i_j}, which can contain:

- Candidate pillar labels and evidence spans,
- Candidate budget lines and amounts (possibly per currency),
- Flags indicating uncertainty or missing information.

These results are stored in a scratchpad-like JSON structure.

Final formatting. After all instructions are executed, a final formatting prompt $P_{\text{Format}}(\{\text{result}_I\})$ asks the LLM to consolidate everything into a single, schema-aligned output:

$$\{y, B\} = \text{LLM}(P_{\text{Format}}(\{\text{result}_I\})), \tag{24}$$

where y is the document-level pillar label vector and B contains pillar-level budget allocations, each with a list of supporting evidence spans. The JSON schema includes fields for `pillar_id`, `budget_amount`, `currency`, and `evidence_span_ids`.

G.5 Chunk and Document Aggregation

For the baselines that operate at the chunk level, we aggregate $\{y, B\}$ tuples into document-level outputs as follows:

- **Labels:** a document is assigned pillar j if at least one chunk has $y_j = 1$; we also report per-pillar coverage (fraction of chunks tagged with each pillar).
- **Budgets:** for each pillar, we sum chunk-level budgets B across all chunks that include that pillar; overlapping allocations (chunks with multiple pillars) are split proportionally based on the model's confidence scores when available, or uniformly otherwise.
- **Conservation:** we compare the sum of all pillar-level budgets against the document's total EWS budget (when annotated) and report conservation error metrics in Section 5.

H Document-Level Aggregation of Chunk Predictions

For evaluation, we aggregate chunk-level outputs to obtain document-level budgets and labels. Let C_d denote the set of chunks associated with document d, and let $B_c \in \mathbb{R}^5_{\geq 0}$ be the pillar-wise budget vector predicted for chunk $c \in C_d$ (missing pillars are treated as zero). The predicted budget for pillar p in document d is

$$\hat{b}_{d,p} = \sum_{c \in C_d} B_{c,p},$$

and the corresponding pillar indicator is

$$\hat{y}_{d,p} = [\![\hat{b}_{d,p} > 0]\!].$$

This simple summation scheme is applied uniformly across all methods (Zero-Shot, Few-Shot, Transformer, Few-Shot-CoT, and Agent), ensuring a consistent mapping from chunk-level predictions to document-level budget vectors $\hat{\mathbf{b}}_d$ and label sets $\hat{y}_{d,p}$.

I Black-Box Assistants: Setup and Additional Results

I.1 Expert-Annotated MDB Evidence Set

The MDB evidence set used in Section 5.2 is derived from a subset of CREWS-related MDB project documents. For each document, domain experts annotated:

- Evidence segments (text or table fragments) that support EWS-relevant budgets,
- The corresponding EWS pillar label(s) for each segment,
- The budget amount assigned to that pillar (normalized to a common currency),
- The document's total EWS budget.

These annotations define the gold evidence–pillar–amount triples and document-level totals against which all systems are evaluated.

I.2 Prompt Design for Gemini 2.5 Flash and OpenAI Assistants

Both Gemini 2.5 Flash and OpenAI Assistants are queried in a single end-to-end pass per document, using prompts that follow the same structure:

1. **Role and scope:** The model is instructed to act as a financial analyst specialized in EWS

and MDB climate adaptation projects.

2. **Task description:** Identify EWS-relevant components, assign them to the five EWS pillars, and extract associated budget amounts.
3. **EWS taxonomy:** A concise description of the five pillars (aligned with Appendix D) and examples of typical activities per pillar.
4. **Methodical instructions:** Stepwise guidance on reading the PDF (narrative, tables, footnotes), checking consistency, and avoiding double counting.
5. **Output schema:** A JSON template requiring, for each document, (i) pillar-level labels and budgets, (ii) a list of evidence segments per pillar, and (iii) a total EWS budget estimate.

Gemini 2.5 Flash receives the full PDF via its native file interface; OpenAI Assistants receive the same content as pre-processed text and tables. Minor token-length adaptations aside, both prompts share the same structure and schema.

I.3 Balanced Evidence Subsample

To test robustness to label imbalance, we construct a balanced subsample of the MDB evidence set with approximately equal support for each EWS pillar. The sampling procedure:

- Identifies the minimum per-pillar evidence count across the full set,
- Uniformly samples that number of evidence segments per pillar,
- Retains only documents that still contain at least one segment for each pillar after sampling.

We recompute all metrics from Section 5.2 on this balanced subset. The qualitative pattern remains unchanged: the Glass-Box Agent maintains the highest macro-averaged scores on evidence extraction, pillar labeling, and pillar-level budget fidelity, with Gemini 2.5 Flash consistently second and OpenAI Assistants third.

I.4 Metric Computation Details

All metrics in Section 5.2 reuse the definitions from Section 5.1 and Appendix G.5:

- TP/FP/FN counts for evidence segments are computed at the segment level (exact-match or strict overlap, depending on annotation granularity).
- Pillar-level budgets $\hat{b}_{d,p}$ for black-box systems are obtained by aggregating their own evidence-level outputs using the same summation rule as the Glass-Box Agent.
- Total-amount accuracy $\mathrm{acc}_{\mathrm{tot}}(d)$ is computed exactly as in Section 5.2, without renormalization of $\hat{B}_d^{\mathrm{tot}}$.

Full numeric tables corresponding to Figures 2 and 3 are provided in the supplementary material.

System	$\mathbf{F_{1\,ev}}$	$\mathbf{F_{1\,pill}}$	$\mathbf{F_{1\,bud}}$	**med.** $\mathrm{acc}_{\mathrm{tot}}$
Glass-Box Agent	0.83	0.82	0.80	0.79
Gemini 2.5 Flash	0.78	0.76	0.74	0.74
OpenAI Assistants	0.67	0.65	0.63	0.64

Table 5: Balanced MDB evidence subsample (approximately equal support per pillar). $F_{1\,\mathrm{ev}}$ = macro F_1 for evidence extraction; $F_{1\,\mathrm{pill}}$ = macro F_1 for pillar labels; $F_{1\,\mathrm{bud}}$ = macro F_1 for pillar-level budget fidelity under the $\pm 5\%$ tolerance band; med. $\mathrm{acc}_{\mathrm{tot}}$ = median total-amount conservation accuracy as in Section 5.2. Values mirror the qualitative pattern in Section 5.3, with the Glass-Box Agent performing best, Gemini 2.5 Flash second, and OpenAI Assistants third.

Variant	Evid. F_1	R@5	Pillar F_1	$\mathrm{Acc}_{\mathrm{tot}}$
Full Agent	0.78	0.86	0.81	0.79
w/o ctx augmentation	0.73	0.82	0.77	0.74
Dense-only retrieval	0.69	0.78	0.74	0.71
$k = 3$ (R@3)	0.72	0.80	0.76	0.75
$k = 10$ (R@10)	0.74	0.84	0.78	0.76
w/o self-healing	0.71	0.82	0.75	0.72

Table 6: Ablation results for the Glass-Box Agent on the MDB evidence development set. Each variant removes or modifies a single component of the full system.

J Ablation Studies

We evaluate four variants of the Glass-Box Agent on the MDB evidence development set, each obtained by removing or modifying one component at a time. Table 6 reports evidence-extraction F_1, Recall@5, pillar-level macro-F_1, and document-level total-amount accuracy (as defined in Section 5.2).

K Cross-Fund Generalization

To assess generalization beyond CREWS Fund documents, we conducted preliminary experiments on a small held-out set of documents from:

- Green Climate Fund (GCF): 15 project documents
- Adaptation Fund (AF): 10 project documents

Without any fine-tuning or re-calibration, we observed the following performance degradation compared to CREWS Fund documents:

The performance drop is primarily attributed to:

1. Different document layouts and table formats

Fund	Accuracy	Precision	Recall
CREWS (in-domain)	0.87	0.89	0.83
GCF (out-of-domain)	0.72	0.75	0.69
AF (out-of-domain)	0.68	0.71	0.65

Table 7: Generalization performance on out-of-domain climate fund documents.

2. Varying terminology for similar EWS activities
3. Different budget reporting conventions

We recommend re-calibrating the system with a small number of labeled examples from the target fund before deployment. Future work will focus on domain adaptation techniques to improve zero-shot generalization.

Limitations

While our approach demonstrates significant improvements in automating financial tracking for EWS investments, several limitations remain. First, our system relies on existing financial reports from MDBs, in this case CREWS, which are often heterogeneous and may contain incomplete or ambiguous financial allocations. In cases where funding details are missing or inconsistently reported, even advanced retrieval-augmented generation (RAG) and multi-step reasoning approaches may struggle to provide accurate classifications. Second, the classification system is influenced by the training data used in fine-tuning and prompt engineering. Despite expert annotations, the model may still exhibit biases in investment classification, particularly when encountering novel financial structures or terminology not well-represented in the dataset (see Section 6 for our mitigation strategies). Third, while our agent-based RAG system achieves state-of-the-art performance on structured and unstructured financial data, its generalizability to other climate finance applications outside EWS has not been fully explored (see Appendix K for preliminary cross-fund results). Future work should assess model robustness across different sustainability reporting frameworks and financial instruments. Fourth, our annotated corpora are modest in size compared to large-scale NLP benchmarks, reflecting the difficulty of obtaining expert-labelled MDB financial data. We mitigate this by using real-world, heterogeneous project reports, document-level splits to avoid leakage, and complementary evaluations at both pillar and evidence level, but broader statistical conclusions will require expanded datasets in future work. Finally, our system assumes that financial tracking can be improved through AI-assisted reasoning; however, its real-world effectiveness depends on institutional adoption, policy integration, and alignment with evolving financial disclosure regulations.

Ethics Statement

Human Annotation. This study relies on annotations provided by domain experts from the WMO, who possess extensive knowledge of Early Warning Systems (EWS). These experts played a pivotal role in the design and conceptualization of the study. Their deep understanding of both the contextual and practical aspects of the collected data ensures the accuracy and relevance of the annotations. The use of expert annotations minimizes the risk of misclassification and enhances the reliability of the model's outputs.

Responsible AI Use. This tool is intended as an assistive system to enhance transparency and efficiency in financial tracking, not as a replacement for human analysts. Expert oversight remains crucial in interpreting financial classifications, addressing edge cases, and ensuring compliance with policy frameworks. By open-sourcing our dataset and model, we encourage responsible use and further validation to refine the system's applicability in real-world climate finance decision-making.

Data Privacy and Bias. This study does not involve any personally identifiable or sensitive financial data. All data used in this research originates from publicly available sources under a Creative Commons license, ensuring compliance with data privacy regulations. While we find no evidence of demographic biases in the dataset, we acknowledge that financial reporting by multilateral development banks (MDBs) may reflect institutional biases in investment classification. Our model operates as a decision-support tool and should not replace human judgment in financial tracking and policy decisions.

Reproducibility Statement. To ensure full reproducibility, we will release all PDFs, codes, EWS-taxonomy, and expert-annotated data used in this study. Our approach aligns with best practices in AI transparency and responsible research dissemination. However, we encourage users of this dataset and model to consider ethical implications when applying automated financial tracking systems in real-world decision-making contexts. For vector database storage and retrieval, we utilized Weaviate, an open-source, scalable vector search engine that efficiently indexes high-dimensional embeddings. Additionally, for reasoning and large language model (LLM) interactions, we integrated OpenAI's API, leveraging its advanced capabilities to process, analyze, and infer patterns from financial document data.

Disclaimer

Opinions expressed in this article are the author's opinions and do not necessarily reflect those of WMO or its Members.

Acknowledgements

This paper has partially received funding from the Swiss National Science Foundation (SNSF) under the project 'How sustainable is sustainable finance? Impact evaluation and automated greenwashing detection' (Grant Agreement No. 100018_207800).