Literature-based extraction of compartmental epidemiology models with MIRA

Rushali Mohbe
Northeastern University
Boston, Massachusetts, USA
mohbe.r@northeastern.edu

Tenzin Nanglo
Northeastern University
Boston, Massachusetts, USA
t.nanglo@northeastern.edu

Klas Karis
Northeastern University
Boston, Massachusetts, USA
k.karis@northeastern.edu

Benjamin M. Gyori
Northeastern University
Boston, Massachusetts, USA
b.gyori@northeastern.edu

Abstract

Epidemiological models are essential to public health decision making, yet they are inaccessible for systematic reuse and repurposing as they are locked in scientific literature as unstructured text and equations. We introduce MIRA, an abstract model representation framework that supports programmatic construction and manipulation of models via ontology-grounded process templates. Building on MIRA, we present MIRA-DB which implements an automated pipeline for extracting compartmental epidemiology models from scientific literature and populates a queriable database exposed through a public web interface. MIRA-DB incorporates clients for literature acquisition and multiple complementary methods for extracting equation content from publications. Extracted equations are parsed into symbolic form using pre-trained language models, assembled into MIRA Template Models, and grounded in domain ontologies, ensuring standardized and comparable model representations across the database. An initial prototype of MIRA-DB contains 559 models and is available at https://epimodels.io.

CCS Concepts

• **Applied computing** → **Computational biology**; **Health informatics**; • **Computing methodologies** → *Information extraction*; *Natural language processing*; *Knowledge representation and reasoning*; • **Information systems** → **Relational database model**; Web interfaces.

Keywords

Epidemiological modeling; knowledge representation; large language models; compartmental models; ontologies

ACM Reference Format:
Rushali Mohbe, Tenzin Nanglo, Klas Karis, and Benjamin M. Gyori. 2026. Literature-based extraction of compartmental epidemiology models with MIRA. In *Proceedings of Proceedings of the epiDAMIK Workshop at KDD 2026 (epiDAMIK '26).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

epiDAMIK '26, Jeju, Korea

ACM ISBN 978-1-4503-XXXX-X/2026/08
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Epidemiological models have been essential in shaping public health responses to infectious diseases, from Ebola outbreak response to HIV/AIDS prevention planning. Compartmental models, widely used in epidemiology, partition a population into discrete compartments (e.g., susceptible, infected, recovered) and describe the flow of individuals between them. They are most often implemented as systems of ordinary differential equations (ODEs) that describe how each compartment changes over time as a function of other compartment states and a set of parameters. However, despite a rich pool of epidemiological modeling literature, these compartmental models remain computationally inaccessible. Model equations are embedded in publication text or figures, their implementations vary across codebases, and their naming conventions can differ across authors. Key modeling assumptions are often not made explicit and are not apparent from equation-level representations.

These barriers make existing models difficult to find, reproduce, and compare, and prevent broader meta-analysis across the modeling literature. In the context of the COVID-19 pandemic, [14] advocated for reproducible model sharing and created a repository of models manually reimplemented from publications, made available in the SBML standard model representation [4] on BioModels.org [9]. However, this repository is limited to 28 models, all specific to COVID-19, and does not provide a general and scalable framework overcoming the above limitations.

To address these limitations, here we present the modeling framework MIRA and the corresponding model database MIRA-DB. MIRA makes three key contributions: it provides (i) a standardized model representation using high-level "Templates" that represent processes appearing in compartmental models, abstracted away from specific mathematical or software implementation, (ii) operations on "Template Models" for extension, stratification, comparison and composition, and (iii) generation and export of model implementations as differential equations and other output formats. Building on MIRA, we introduce MIRA-DB, which implements scalable pipelines for literature access, text and image processing, extraction of model equations, ontology grounding of extracted concepts, and a database schema and web service that stores models with their provenance and metadata. We benchmarked multiple publication

processing and model extraction approaches against a set of curated gold-standard models. Benchmarks showed that some extraction methods reached high performance, matching gold-standard equations above 92% in shared equation terms, and revealed trade-offs between cost and accuracy.

Both MIRA[1] and MIRA-DB[2] are available on GitHub under permissive licenses. An initial prototype of MIRA-DB comprising 559 models is available as a public web service at https://epimodels.io.

2 Model representation through ontology-grounded templates

We introduce MIRA, an abstract model representation framework for compartmental models that captures the high-level semantics of modeling processes while abstracting away from specific mathematical or software implementations. MIRA does this by introducing domain-independent structural templates such as *NaturalDegradation* and *ControlledConversion* that can represent typical patterns of compartmental processes in epidemiology and other fields (systems biology, ecology, etc.). Templates take one or more *Concept*s as arguments that represent compartments, where each *Concept* is identified by a term in a domain ontology that determines its meaning independent of the naming conventions used in any particular publication. For epidemiological modeling, MIRA draws on ontologies such as the Infectious Disease Ontology (IDO) [1] and the Apollo Structured Vocabulary [3]. A MIRA template-based representation of a simple Susceptible-Infectious-Recovered (SIR) compartmental model is shown in Figure 1. MIRA Templates can further capture custom rate laws using SymPy expressions corresponding to the process represented. A MIRA Template Model then consists of a collection of Templates, a list of Parameters (which can also be given ontology-grounding and can appear in Template rate laws), Initial Conditions (expressions that define starting values of Concepts), Observables (expressions over Concepts and Parameters defining a model readout) as well as several attributes that carry provenance and metadata associated with the model. The MIRA Template Model and all model components are implemented as Python classes allowing for the structured, programmatic construction of models, an approach inspired by frameworks developed for systems biology [2, 8, 15].

MIRA provides a suite of modeling operations on Template Models including model extension, model stratification (e.g., stratifying compartments of an SIR model into multiple age groups or cities), model comparison and model composition. Template Models can also be exported to other modeling formalisms, including Petri net representations [6] and community exchange formats such as SBML [4].

3 Automated extraction of models from publications

Building on MIRA as a modeling framework, the purpose of MIRA-DB is to implement a scalable pipeline for the extraction of model implementations from publications as MIRA Template Models, and to create a queriable model database. The MIRA-DB pipeline consists of four stages: (i) finding, acquisition and filtering of relevant publications, (ii) publication processing and language model-based extraction of model equations, (iii) mapping equations to ontology-grounded MIRA Templates, and (iv) structured storage of models and metadata in a relational database.

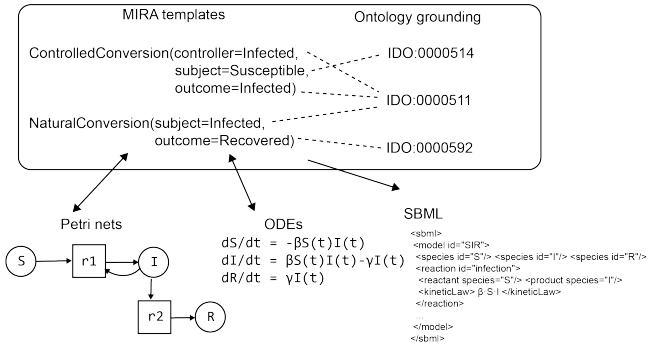


Figure 1: MIRA Template representation of a basic SIR model with ontology-grounded templates (rounded box). Arrows show import/export modalities into common modeling formalisms.

For publication acquisition, MIRA-DB implements an interface toward the PubMed API [16], which provides broad coverage of metadata on publications in epidemiology, biology, medicine, and related fields, and PubMedCentral, which provides access to full-text content for a large portion of PubMed entries. A relevant input corpus of publications is defined via a custom combination of topic terms and keywords passed to the PubMed API. Because compartmental models are typically described only in publication full text (rather than the abstract), MIRA-DB takes full text content as input. In our prototype, this is sourced through PubMedCentral, though other sources of text content can also be used as input to MIRA-DB.

3.1 Extracting equation content from publication text

MIRA-DB implements four complementary methods that take a publication as input and return a textual or image representation of equations defining a compartmental model (if such model exists in the publication). This is challenging because model equation content must be reliably isolated from the broader document structure, which typically contains multiple tables, figures, and text in custom formats. The four methods incorporated in MIRA-DB differ in the input format taken, the output format produced, and the methodology of extraction, and thus vary in extraction performance and computational cost.

PubMedCentral makes full text publications available in several formats, including a custom XML schema and PDF, though the availability of a specific format differs across papers. The first approach ("XML-Markup") implemented by MIRA-DB traverses the XML-formatted version of a publication (when available) to identify tags encompassing MathML or LaTeX-formatted equations and returns these as text. The other extraction approaches take a PDF as input and wrap the Marker [11] or MinerU [17] libraries for PDF processing.

[1] https://github.com/gyorilab/mira
[2] https://github.com/gyorilab/miradb

The Marker library converts PDFs to structured HTML from which equation content is extracted. A benefit of Marker is that its output is structured, making it suitable for systematic parsing. We call this extraction method "Marker-HTML".

MinerU is a Python package that extracts text content and images from PDFs as separate output artifacts. It identifies equations within a PDF and outputs these both as text ("MinerU-Text") and as images ("MinerU-Image").

3.2 Parsing symbolic equations and mapping to MIRA Templates

Extraction provides a textual or image representation of model-defining equations from a publication. However, these are unstructured formats that require parsing into a symbolic form to be interpretable for model construction. We use the SymPy [10] library for symbolic equation representation and implemented a large language model (LLM)-based approach for generating such representations from raw text or image input. The LLM-based processing pipeline consists of three stages: initial symbolic equation conversion, error correction feedback, and grounding to ontology terms.

First, the system of equations (either as text or image) is provided within a templated prompt to an LLM as input. The prompt provides a general description and examples for the conversion task and instructs the model to return SymPy code declaring the time-dependent state variables and parameters, and the system of equations over these variables. To ensure structural validity, a feedback loop dynamically executes the LLM-generated SymPy code. If a runtime error occurs, the equations are sent back to the LLM for correction using a second prompt. Finally, after a valid system of symbolic equations is obtained, state variables are grounded against domain ontology terms. The LLM is prompted using a template containing the SymPy equations and detailed instructions and examples of concept groundings (from a set of manually curated epidemiology models) to guide this grounding step.

Given a symbolic equation representation, the equations are assembled into a MIRA Template Model using a hypergraph-based algorithm over the equation terms (see Appendix).

4 Evaluation of automated model extraction

We benchmarked MIRA-DB's extraction pipeline against a set of manually curated gold standard reference models. For each publication in the gold standard, we scored the Template Model produced by each of the four extraction methods against the hand-curated reference model using a three-component structural similarity metric. The resulting scores allow for a direct comparison of extraction quality across methods and document formats.

To create the gold standard model set, we chose 12 papers from the BioModels database epidemiology model subset. For each paper, a Template Model was manually curated to accurately represent the published equations, compartment structure, and dynamics, forming the gold standard against which all four extraction methods were evaluated.

Comparing automatically extracted ODEs to a gold standard is non-trivial since the same mathematical structure can be expressed with many notational variants and equivalent dynamics can be written in algebraically different but mathematically identical forms.

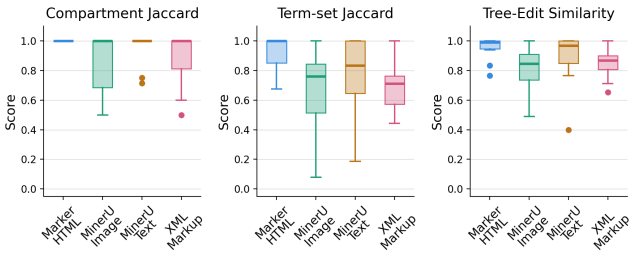


Figure 2: Score distributions by extraction method. Box plots show Compartment Jaccard, Term-set Jaccard, and Tree-edit similarity (TES) for each of the four extraction methods across all evaluated papers.

Simple string or token matching fails to capture structural equivalence. We therefore defined a three-component structural similarity metric that operates at increasing levels of granularity: the *Compartment Jaccard* (CJ) measures overlap in compartment variable sets after fuzzy name matching, the *Term-set Jaccard* (TJ) measures overlap in canonicalized symbolic terms on the right-hand side of each ODE, and the *Tree-Edit Similarity* (TES) measures fine-grained compositional similarity between equation expressions as abstract syntax trees. A combined similarity score weights these three components, with CJ acting as a gating signal so that TJ and TES contributions are scaled by compartment overlap. Full definitions and the combined score formula are given in Appendix 7.

Figure 2 shows score distributions across the three metric components for each extraction method; the same results are reported in numerical form in Appendix Table 1. Marker-HTML consistently outperforms the other methods, achieving a median Compartment Jaccard of 1.0 with near-zero variance and the highest Term-set Jaccard scores, indicating reliable recovery of both state variables and equation structure. MinerU-Text equals Marker on compartment recovery but shows high variance in Term-set Jaccard, likely due to optical character recognition errors in Greek letters and subscripted variables. MinerU-Image and XML-Markup perform similarly to each other, with moderate scores across both Jaccard components. TES scores are consistently high across all methods, suggesting that when terms are correctly identified, their compositional structure is generally preserved.

Appendix Table 2 shows the combined scores per paper across all four extraction methods. Marker-HTML consistently achieves the highest combined score across most papers, with scores above 0.9 for the majority of the gold standard. Performance varies notably across papers rather than uniformly across methods. For example, PMID 32703315 is a low scoring outlier for MinerU-Image, while MinerU-Text drops sharply for PMID 32341628 despite other methods performing well. This per-paper variation suggests that document-specific factors, such as equation formatting, style, and PDF rendering quality, play a significant role in extraction performance alongside method choice. For instance, a limitation of MinerU-Text is that it often misrecognizes Greek letters and variables with superscripts or subscripts, which are pervasive in compartmental epidemiology models.

5 Constructing a database of literature-extracted epidemiology models

MIRA-DB schema and design considerations. The MIRA-DB schema captures the full chain from source publication to grounded Template Model across four tables (see Appendix Figure 1). The *Text Reference* table stores bibliographic metadata (PubMed ID, DOI, authors, journal, etc.) for each publication. The *Text Content* table records the artifact (text or image) produced by applying a specific extraction method to a publication; a single publication may have multiple *Text Content* rows, one per extraction method applied. The *ODE Expressions* table holds the SymPy-parsed equations derived from a given extracted artifact, including any corrections produced by the LLM error-correction loop. The *MIRA Template Models* table stores the final grounded Template Model serialized as JSON, along with ontology grounding metadata, linked to the originating ODE expression.

Two design choices make the schema extensible. First, the extraction method is encoded as an enumerated type rather than free text, so additional extraction pipelines can be added without schema migration. Second, the one-to-many chain from *Text Reference* through to *MIRA Template Models* allows multiple extraction methods, multiple equation parses, and multiple resulting Template Models to coexist for a single publication, supporting head-to-head comparison and downstream analysis by method.

MIRA-DB content and interface. The current MIRA-DB content was sourced from a PubMed query using the topic term "Epidemiology" combined with keyword constraints matching "compartmental model", "SEIR model", or "SIR model". This query retrieved 1, 953 publications, of which 1, 075 had full text available (55%); the extraction pipeline recovered equations and produced a grounded MIRA Template Model for 559 of these (52%). Models in MIRA-DB show a broad distribution of sizes in terms of the number of Concepts (i.e., compartments) and number of Templates (Appendix Figure 2) with the largest model from [5] containing 16 compartments corresponding to granular disease stages of COVID-19.

We found that both the equation extraction methods and the downstream LLM processing stage had substantially different compute requirements and runtimes, with the LLM stage notably slower for image-based input (see Appendix).

MIRA-DB exposes the extracted models through a web interface that supports model search based on metadata, publication titles, and ontology-grounded concepts. It also renders equations corresponding to the ODE export of the extracted Template Model, and provides download options as Template Model JSON, SBML, and SymPy code.

6 Conclusion and future work

We presented MIRA, an abstract model representation framework for compartmental models, and MIRA-DB, an automated pipeline and database for extracting models from scientific literature into the MIRA representation. MIRA contributes a domain-independent template-based representation in which compartments, parameters, and processes are grounded against domain ontologies, enabling standardized, machine-comparable model semantics together with modeling operations such as stratification, comparison, and composition. MIRA-DB contributes a scalable extraction pipeline that combines complementary methods for recovering equation content from publication full text, parses equations into symbolic form using pre-trained language models, and assembles ontology-grounded MIRA Template Models stored in a queriable database with a public web interface. Together, MIRA and MIRA-DB address a practical bottleneck for the field, the absence of machine-readable, comparable epidemiological models at scale, which has limited model reuse, cross-study meta-analysis, similarity-based retrieval, and the application of foundation models to compartmental epidemiology.

A current limitation is that the evaluation does not yet include a root-cause analysis of extraction method differences. It is unclear whether performance variation stems from document structure, equation complexity, optical character recognition quality, or LLM behavior during extraction. To this end, we plan to expand the manually curated gold standard to cover a broader range of diseases and model types, and to conduct a systematic analysis of extraction failure modes. Another limitation of the current MIRA-DB pipeline is that it focuses on model structure. Although parameter symbols are extracted, extending the pipeline to also recover parameter *values* (leveraging e.g., [7]) is an important next step toward fully reusable models.

To grow MIRA-DB beyond the prototype scope, we plan to scale up the literature corpus and broaden publication sources beyond PubMed. Scaling the literature scope beyond targeted keyword queries will require automated paper relevance classification; in preliminary work, we developed a classifier using document embeddings fine-tuned via contrastive learning on a manually curated set of positive and negative examples. Beyond paper-derived models, the same MIRA representation can capture models implemented in software, suggesting a complementary extraction pathway directly from model source code [13]. The MIRA framework also generalizes beyond compartmental ODE models, both to other domains where similar process-template structures are prevalent (e.g., systems biology, pharmacokinetics) and to other model formalisms.

We also plan to extend MIRA-DB's accessibility and downstream utility. Programmatic access through an API and a Model Context Protocol (MCP) server would enable agents and analysis pipelines to query the database directly. Integration with simulation frameworks [12] would allow extracted models to be parameterized and run without manual reimplementation. The structured model corpus also enables data-driven studies of the modeling literature, including model clustering, cross-disease structural comparison, and fine-tuning language models on the corpus to support tasks such as model synthesis, summarization, and similarity-based retrieval.

7 Acknowledgments

This work was funded under the DARPA Automating Scientific Knowledge Extraction and Modeling (ASKE-M) program (ARO grant number HR00112220036). We thank Abby Leung and Samuel V. Scarpino for helpful comments and discussion.

References

[1] Shane Babcock, John Beverley, Lindsay G Cowell, and Barry Smith. 2021. The infectious disease ontology in the age of COVID-19. *Journal of biomedical semantics* 12, 1 (2021), 13.

[2] Benjamin M Gyori, John A Bachman, Kartik Subramanian, Jeremy L Muhlich, Lucian Galescu, and Peter K Sorger. 2017. From word models to executable

models of signaling networks using automated assembly. *Molecular systems biology* 13, 11 (2017), MSB177651.
[3] William R Hogan, Michael M Wagner, Mathias Brochhausen, John Levander, Shawn T Brown, Nicholas Millett, Jay DePasse, and Josh Hanna. 2016. The Apollo Structured Vocabulary: an OWL2 ontology of phenomena in infectious disease epidemiology and population biology for use in epidemic simulation. *Journal of biomedical semantics* 7, 1 (2016), 50.
[4] Sarah M Keating, Dagmar Waltemath, Matthias König, Fengkai Zhang, Andreas Dräger, Claudine Chaouiya, Frank T Bergmann, Andrew Finney, Colin S Gillespie, Tomáš Helikar, et al. 2020. SBML Level 3: an extensible format for the exchange and reuse of biological models. *Molecular systems biology* 16, 8 (2020), MSB199110.
[5] Françoise Kemp, Daniele Proverbio, Atte Aalto, Laurent Mombaerts, Aymeric Fouquier d'Hérouël, Andreas Husch, Christophe Ley, Jorge Gonçalves, Alexander Skupin, and Stefano Magni. 2021. Modelling COVID-19 dynamics and potential for herd immunity by vaccination in Austria, Luxembourg and Sweden. *Journal of Theoretical Biology* 530 (2021), 110874.
[6] Sophie Libkind, Andrew Baas, Micah Halter, Evan Patterson, and James P Fairbanks. 2022. An algebraic framework for structured epidemic modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 380, 2233 (2022).
[7] Chunwei Liu, Enrique Noriega-Atala, Adarsh Pyarelal, Clayton T Morrison, and Mike Cafarella. 2025. Variable Extraction for Model Recovery in Scientific Literature. In *Proceedings of the 1st Workshop on AI and Scientific Discovery: Directions and Opportunities*, Peter Jansen, Bhavana Dalvi Mishra, Harsh Trivedi, Bodhisattwa Prasad Majumder, Tom Hope, Tushar Khot, Doug Downey, and Eric Horvitz (Eds.). Association for Computational Linguistics, Albuquerque, New Mexico, USA, 1–12. doi:10.18653/v1/2025.aisd-main.1
[8] Carlos F Lopez, Jeremy L Muhlich, John A Bachman, and Peter K Sorger. 2013. Programming biological models in Python using PySB. *Molecular systems biology* 9 (2013), 646.
[9] Rahuman S Malik-Sheriff, Mihai Glont, Tung VN Nguyen, Krishna Tiwari, Matthew G Roberts, Ashley Xavier, Manh T Vu, Jinghao Men, Matthieu Maire, Sarubini Kananathan, et al. 2020. BioModels—15 years of sharing computational models in life science. *Nucleic acids research* 48, D1 (2020), D407–D415.
[10] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. 2017. SymPy: Symbolic Computing in Python. *PeerJ Computer Science* 3 (2017), e103. doi:10.7717/peerj-cs.103
[11] Vik Paruchuri. 2026. *Marker: Convert PDF to Markdown and JSON Quickly with High Accuracy.* https://github.com/datalab-to/marker Accessed: May 19, 2026.
[12] Joshua L Proctor and Guillaume Chabot-Couture. 2024. Democratizing infectious disease modeling: an AI assistant for generating, simulating, and analyzing dynamic models. *medRxiv* (2024), 2024–07.
[13] Adarsh Pyarelal, Marco Antonio Valenzuela-Escárcega, Rebecca Sharp, Paul Douglas Hein, Jon Stephens, Pratik Bhandari, HeuiChan Lim, Saumya Debray, and Clayton T. Morrison. 2020. AutoMATES: Automated Model Assembly from Text, Equations, and Software. *arXiv* abs/2001.07295 (2020). arXiv:2001.07295 https://arxiv.org/abs/2001.07295
[14] Kausthubh Ramachandran, Matthias König, Martin Scharm, Tung VN Nguyen, Henning Hermjakob, Dagmar Waltemath, and Rahuman S Malik Sheriff. 2022. FAIR Sharing of Reproducible Models of Epidemic and Pandemic Forecast. (2022). https://doi.org/10.20944/preprints202206.0137.v1
[15] Adrien Rougny, Vasundra Touré, John Albanese, Dagmar Waltemath, Denis Shirshov, Anatoly Sorokin, Gary D Bader, Michael L Blinov, and Alexander Mazein. 2021. SBGN Bricks Ontology as a tool to describe recurring concepts in molecular networks. *Briefings in bioinformatics* 22, 5 (2021), bbab049.
[16] Eric Sayers. 2022. A General Introduction to the E-utilities. In *Entrez Programming Utilities Help*. National Center for Biotechnology Information (US), Bethesda, MD. https://www.ncbi.nlm.nih.gov/books/NBK25497/ Updated November 17, 2022.
[17] Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, et al. 2024. Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839* (2024).

Appendix

Hypergraph-based mapping of symbolic equations to MIRA Templates

Given a symbolic equation representation of a model, we construct a MIRA Template Model representation by recognizing collections of terms on the right-hand side of ODEs that together correspond to a high-level template such as "ControlledConversion". We developed a hypergraph-based algorithm in which each term on the ODE right-hand side becomes a node. Groups of terms whose symbolic sum is zero are connected via a directed hyperedge from the negated (consumed) terms to the positive (produced) terms; overlapping hyperedges that share terms are resolved by greedy matching so each term belongs to at most one. Each hyperedge is then mapped to a conversion-type template (e.g., NaturalConversion or ControlledConversion), while terms not covered by any hyperedge are mapped to production or degradation templates. Compartments are identified as the left-hand side state variables and parameters as the remaining free symbols. Together with the grounding information obtained from the prior LLM call, the assembled templates form a MIRA Template Model.

Structural similarity metrics for equation extraction benchmarking

The *Compartment Jaccard* (CJ) metric measures the overlap in compartment variable sets between two sets of equations after fuzzy name matching to handle notation differences (e.g., matching S_r to Sr). It is the coarsest signal: two models that share all compartments may still differ dynamically, but models with low CJ are structurally dissimilar by definition. The *Term-set Jaccard* (TJ) metric operates on the symbolic content of each ODE. Each equation's right-hand side is canonicalized using SymPy by stripping scalar coefficients and normalizing and simplifying the terms; the resulting term sets are then compared across corresponding equations. TJ captures whether two models contain similar epidemiological interactions independent of parameter naming conventions. The *Tree-Edit Similarity* (TES) metric measures the fine-grained structural similarity of individual equation expressions as abstract syntax trees and captures compositional similarities even when term sets match. TES scores are normalized to $[0, 1]$, with 1 indicating identical structure.

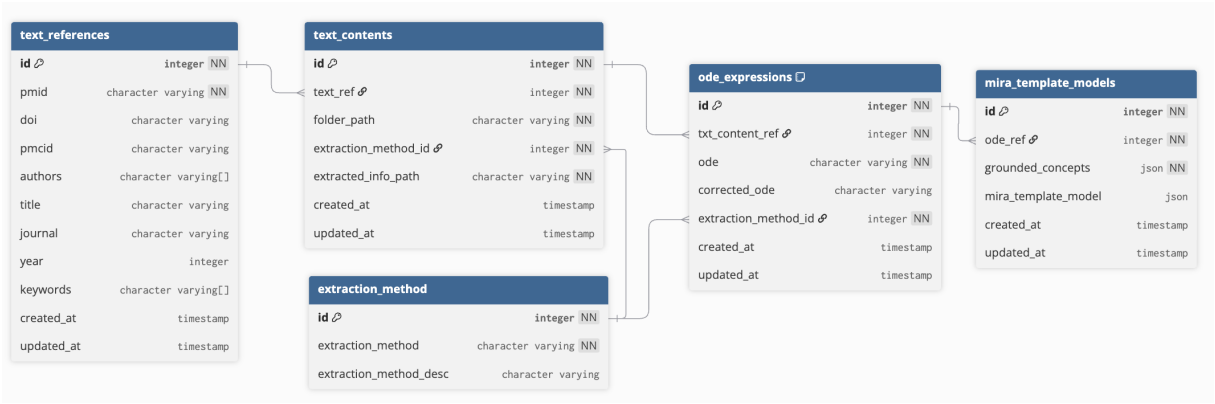
The combined similarity score combines the three components:

$$\mathrm{S}_{\mathrm{combined}} = 0.2 \cdot CJ + 0.5 \cdot CJ \cdot TJ + 0.3 \cdot CJ \cdot TES$$

CJ acts as a gating signal: models with little compartment overlap are structurally incomparable, so TJ and TES contributions are scaled by CJ. TJ receives the highest weight (0.5) because matching symbolic terms most directly captures similarity in model dynamics. TES (0.3) provides finer-grained differentiation when models share terms but differ in composition. The standalone CJ term (0.2) assigns partial credit for shared compartment structure even if equation-level similarity is low.

Compute requirements for equation and model extraction

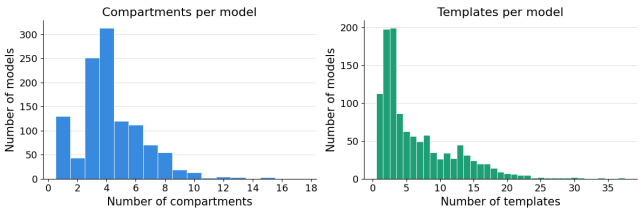
The compute requirements for equation extraction from publications differed substantially depending on the method. Marker took



Appendix Figure 1: Schema of the MIRA-DB database, each table shown as a rectangle with table name in the header and attributes as rows. Relationships between tables are shown using edges. The figure was created using dbdiagram.io.

on average 4 minutes 20 seconds per publication on a T4 Tensor Core GPU, whereas MinerU took on average 4 minutes on an Apple M2 Pro CPU. Given the simple XML traversal involved, the time for XML-Markup extraction is negligible.

For the LLM-based processing stage of model extraction, we used the gpt-4o-mini model with default parameters across all LLM calls. On average, each paper required 10,407 input tokens (including the fixed prompt cache) and 1,348 output tokens, at an average cost of $0.002 per paper. LLM processing time averaged 21 seconds for text-based inputs (XML-Markup, Marker-HTML, MinerU-Text) and 1 minute 28 seconds for image-based input (MinerU-Image).



Appendix Figure 2: Distributions of model sizes across MIRA-DB. Left: number of compartments (Concepts) per model. Right: number of Templates per model.

Appendix Table 1: Score distribution by extraction method (mean ± standard deviation across the 12 gold-standard papers). These are the same results visualized in Figure 2.

Method	Compartment Jaccard	Term-set Jaccard	Tree-Edit Similarity
Marker-HTML	1.0 ± 0.0	0.923 ± 0.108	0.955 ± 0.077
MinerU-Image	0.849 ± 0.21	0.67 ± 0.27	0.815 ± 0.15
MinerU-Text	0.955 ± 0.105	0.79 ± 0.251	0.885 ± 0.175
XML-Markup	0.885 ± 0.194	0.694 ± 0.161	0.848 ± 0.105

Appendix Table 2: Per-paper combined similarity scores by extraction method. Dashes indicate papers for which the extraction method did not produce a usable Template Model.

PubMed ID	Marker-HTML	MinerU-Image	MinerU-Text	XML-Markup
32046137	0.982	0.429	0.969	0.433
32219006	0.767	0.636	0.767	—
32289100	1.0	0.89	0.89	0.866
32322102	1.0	0.631	1.0	1.0
32341628	0.941	0.844	0.31	0.856
32574303	1.0	0.843	1.0	0.832
32616574	0.866	0.866	1.0	0.463
32703315	0.888	0.193	0.907	0.861
32706790	1.0	1.0	1.0	0.739
32735581	0.929	0.96	0.888	0.803
32834593	1.0	0.42	0.496	—
32834603	1.0	0.453	0.799	0.359
Mean	0.948	0.68	0.836	0.721