HierLex: Launch & Navigation Guide

What HierLex Does

HierLex is an AI-assisted legal pattern discovery and forensic comparison system designed to identify the factual and legal mechanisms that distinguish successful litigation strategies from unsuccessful ones.

Unlike conventional legal search systems that retrieve cases using keyword matching or citation frequency, HierLex analyzes the substantive legal arguments contained within judicial opinions and groups them into recurring litigation patterns. After the user selects a collection of court decisions, HierLex automatically performs the following analytical workflow:

1. Document Processing

Each uploaded opinion is cleaned and converted into structured text suitable for machine analysis. HierLex preserves the original language of each opinion while preparing the documents for both large language model (LLM) analysis and semantic embedding.

2. Winning and Losing Argument Extraction

HierLex separates opinions according to litigation outcome. For each opinion, advanced Large Language Models (LLMs) identify the principal legal arguments that contributed to the prevailing party's success and, separately, the arguments or deficiencies associated with the losing party. Rather than extracting isolated quotations, HierLex summarizes the underlying causal legal mechanisms contained within each opinion, such as evidentiary sufficiency, procedural compliance, statutory interpretation, constitutional reasoning, administrative procedure, factual causation, burden of proof, and judicial deference. The extracted mechanisms become the fundamental analytical units used throughout the remainder of the pipeline.

3. Automatic Taxonomy Construction

After collecting legal arguments from all cases, HierLex asks the LLM to construct a hierarchical legal taxonomy consisting of Root Patterns (broad categories representing recurring legal mechanisms) and Sub-Patterns (more specific litigation strategies or factual mechanisms contained within each Root Pattern). These taxonomies are automatically generated from the supplied cases rather than being based upon a predefined legal ontology.

4. Winning/Losing Pattern Alignment

Winning and losing arguments rarely use identical language. Accordingly, HierLex asks the LLM to align losing arguments with the existing winning taxonomy. This produces a common analytical framework in which both successful and unsuccessful litigation strategies are organized under identical Root and Sub-Pattern identifiers. Using a shared taxonomy allows direct comparison between opposing litigation mechanisms.

5. Semantic Embedding

Once the taxonomy has been finalized, HierLex converts every Sub-Pattern into numerical semantic vectors using two independent embedding models: LegalBERT, which captures legal-domain semantic relationships learned from judicial opinions, and Jina v2, which provides a second, independently trained embedding space designed for robust semantic retrieval and similarity measurement. Each Sub-Pattern therefore receives two independent semantic representations.

6. Embedding Individual Cases

Each court opinion is converted into semantic vectors using the same embedding models. Long opinions are divided into manageable sections before embedding. The resulting vectors represent the overall semantic content of each opinion while preserving important litigation details.

7. Similarity Analysis and Excel Output

After constructing the legal taxonomy, HierLex compares every court opinion against every Sub-Pattern using semantic similarity. For each Sub-Pattern, HierLex computes similarity scores between that Sub-Pattern and every case in the uploaded dataset. The resulting similarity metrics are first written to *HierLex_Final_Pattern_Analysis.xlsx*, where each worksheet summarizes a different stage of the analysis. In particular, the Case_Pattern_Alignment worksheet contains dedicated columns for each automatically generated Sub-Pattern (e.g., Pattern_1_1, Pattern_1_2, …) and the corresponding similarity

metrics. Each row represents a single court opinion. Consequently, every case is compared against every discovered Sub-Pattern, enabling users to examine how strongly each opinion aligns with every litigation pattern identified by HierLex.

8. Final Pattern Analysis Workbook

All analytical results are exported into *HierLex_Final_Pattern_Analysis.xlsx* at the user-designated output directory. The workbook contains multiple worksheets summarizing different aspects of the analysis, including extracted winning and losing arguments, Root/Sub-Pattern taxonomy, similarity scores, global/differential similarity measures, pattern alignment results, and distinguishable litigation arguments. The accompanying HierLex v1.5 Forensic Analysis Manual explains the purpose of each worksheet and provides detailed guidance on interpreting the reported metrics and similarity scores.

Human Interpretation

HierLex is designed to assist—not replace—legal judgment. The software identifies recurring litigation mechanisms and quantifies their semantic similarity across cases. Final legal interpretation, strategic assessment, and citation decisions remain the responsibility of the attorney or researcher.

1. GUI Navigation & Taxonomy Overview

HierLex is designed for high-fidelity legal corpus analysis. The interface is divided into functional zones to ensure a controlled forensic workflow.

Section	Function
LLM Provider Configuration	Toggles between AI backends. Advanced providers enable higher-fidelity synthesis. Refer to #1. Acquire Credentials under Section 2 Launch & Acquisition Procedure for more details.
API Key Input	Secure field for your digital "ID badge" (authentication token). Refer to #1. Acquire Credentials under Section 2 Launch & Acquisition Procedure for more details.
Directory Setup	Loads your legal PDF/Image corpus for processing from Select Court Cases Directory where your court cases are saved: note that

	each court case should be saved in a separate file name. Court cases in PDF are most preferred. Selects Results Directory where your final result will be saved.
K N Specifications & Model Selection	Sets K (Root Patterns: macro axes) and N (Sub Patterns: tactical variants) for pattern clustering: refer to Understanding the HierLex Taxonomy below this table for more details. Set (K * N) closer to the case number (or less than the case number if the number is much larger than (K * N)). For instance, if you have 12 court cases, set K = 3 and N = 4, with K < N for the best results. If (K * N) exceeds the case number, LLMs might lead to overfitting or hallucinations. For Model Selection, refer to #2. Configure under Section 2. Launch & Acquisition Procedure.
EXECUTE ANALYSIS	Initializes the background worker to begin the forensic extraction process.
INTERRUPT ANALYSIS	Safely halts the background engine mid-process without closing the application.
Log Window	Real-time forensic audit trail of the extraction process.

Understanding the HierLex Taxonomy

HierLex organizes legal arguments into a hierarchical structure to identify where your case stands tactically, leveraging dual-model (LegalBERT/Jina) alignment.

Component	Definition	Strategic Value
Root Pattern	The *Macro-Doctrinal Anchor* (high-level legal theory).	Defines the legal "battlefield."
Sub-Pattern	The *Tactical Signature* (specific evidentiary strategies).	Defines the "winning move."

2. Launch & Acquisition Procedure

HierLex utilizes external AI "engines" for forensic analysis. Think of these as a **metered subscription service** for computational power.

1. **Acquire Credentials:** Visit the AI provider's portal (e.g., OpenAI Platform). Create an account, add a credit card to your "Billing" section, and generate an **"API Key." You should save this Key securely so that nobody else can access it. It is recommended to set a monthly limit to prevent accidental overcharging caused by a potentially endless LLM loop.**
2. **Configure:** Paste this key into the *API Key Input* field in HierLex. It acts as your digital ID badge. When configuring the Model Selection, select a lower-capacity model (e.g., gpt-4o-mini) for the Sub Model field, which handles high-volume per-chunk argument extraction and benefits from speed and cost efficiency. Reserve a higher-capacity model (e.g., gpt-4o) for the Root Model field, which performs the comprehensive cross-case meta-analysis and taxonomy synthesis requiring deeper reasoning and broader contextual understanding. Most major LLM providers — including OpenAI, Anthropic (Claude), Google (Gemini), Perplexity, Grok, Meta, Copilot and others — offer a range of models at varying capacity and cost tiers through their API portals; consult your chosen provider's model documentation to identify the appropriate Sub Model and Root Model pairing for your budget and analytical requirements.
3. **Input:** Load your PDF directory or supported image files via the Directory Setup. Select the file path where your court cases in PDF (most preferred) are saved.
4. **Run:** Click **EXECUTE ANALYSIS** to initiate the background engine. Threaded execution ensures the GUI remains responsive during analysis.
5. **Interrupt:** Click **INTERRUPT ANALYSIS** if you need to stop the engine; the system will gracefully release resources and reset to a ready state.

3. Performance & Resource Expectations

HierLex v1.5 performs intensive "Chain-of-Thought" reasoning and dual-model similarity computations to extract legal patterns using optimized LegalBERT/Jina v2 embeddings. Users should anticipate:

- **Processing Speed:** Highly dependent on case volume; estimated at ~5 minutes per case depending on hardware and token complexity.
- **System Load:** The app uses `spawn`-method multiprocessing. To maintain optimal performance, avoid running other high-CPU applications simultaneously.

> **Operational Tip:** If the **EXECUTE ANALYSIS** button remains inactive, the engine is currently performing an analysis. Monitor the **Log Window** for real-time progress. If a specific case takes longer than expected, it is likely due to an unusually high token count or complex image OCR requirements in that document.