

A guide for lawyers to understanding how LLMs work

Large Language Models are revolutionizing the legal profession, offering transformative potential when used safely and ethically

By Jeffrey I. Ehrlich

Preamble – How this article was written

When I first experienced ChatGPT in late 2022, I wondered, "How does this thing actually work?" Since then, I have been looking for articles or lectures that answer that question. But what I have found has either been pitched at such a high level of generality that it did not really provide a meaningful answer (it's like autocorrect on your phone), or was geared toward a reader with a technical background, which is not me. So, with the help of Grok 3, Claude 4, and ChatGPT 4, three of the most capable large-language models (LLMs), I have created this article, trying to bridge that gap.

The process was more collaborative than you might expect. While I had done enough research to have a reasonable understanding of the topic, that understanding was too general in important spots to allow me to write the article without the assistance of AI. Nor was I able to just ask an LLM to write a suitable article. I tried. They tried. Their product was not usable.

So, I started by explaining to Grok 3 what I was hoping to create, and then Grok and I engaged in an iterative process that ultimately took a few hours before I was reasonably satisfied with the result. As Grok produced drafts, I would ask it to explain some parts more clearly, or I would propose new areas that I wanted to cover, or better examples of concepts I wanted it to discuss.

Then, when the article was finished, I ran it by ChatGPT 4 and Claude 4 Sonnet for good measure, both to try to ensure that the technical aspects that I do not understand were explained accurately, and for suggestions on how to improve the article's text. The article below is the product of that collaborative process.

If you take away anything from this article, it's this: **DO NOT, UNDER ANY CIRCUMSTANCES, RELY ON CASE CITATIONS PROVIDED BY *ANY LLM*, INCLUDING THE ONES OFFERED BY LEXIS AND WESTLAW, UNLESS YOU HAVE PERSONALLY VERIFIED THAT THE CITED CASE EXISTS AND SAYS EXACTLY WHAT YOU ARE CITING IT FOR.**

Introduction

Large language models, such as those powering ChatGPT, Claude, or Grok, are revolutionizing the legal profession by automating tasks like contract drafting, case-law analysis, document review, and e-discovery. These tools offer unmatched efficiency, scalability, and insight, enabling lawyers to process vast datasets, generate legal documents, or identify relevant evidence

Copyright © 2025 by the author.
For reprint permission, contact the publisher: **www.plaintiffmagazine.com**



swiftly. But their "black box" nature raises critical questions about reliability, bias, copyright, and accountability – issues central to legal practice.

The term "black box" refers to systems where we can see inputs (your prompts) and outputs (the LLM's responses), but the internal decision-making process remains opaque. Unlike traditional legal research where you can trace each step – from search query to specific cases to quoted passages – LLMs generate responses through billions of numerical calculations that can't be meaningfully explained or audited. This opacity becomes problematic when an LLM confidently cites a non-existent case or misinterprets a statute, as lawyers cannot trace how or why the error occurred.

LLMs' ability to generate novel responses, their potential to "hallucinate" incorrect information (e.g., nonexistent cases or statutes), and their limitations in tracking information sources demand careful scrutiny. This article seeks to demystify LLMs by explaining their inner workings – tokenization, parameter encoding, attention mechanisms, training, and reinforcement learning – in a clear, accessible manner for legal professionals with minimal technical background. It explores how LLMs generate responses, their practical applications like e-discovery and litigation, and their ethical implications, equipping lawyers to leverage their benefits while mitigating risks like bias, inaccuracy, and breaches of professional responsibility. By understanding these systems, lawyers can use LLMs to enhance their practices while avoiding the risks endemic to their use.

If you want a list of key takeaways for safe use of LLMs in legal practice, I'll provide it here:

- Never trust any citation without verification;
- Use LLMs for first drafts, not final products;
- Best for: brainstorming arguments, organizing thoughts, identifying issues; reviewing and summarizing documents;
- Worst for: finding specific cases, quoting exact language, jurisdictional nuances;
- Always maintain client confidentiality – check the LLM's data-use and retention policies;
- Document your verification process.

The building blocks: Tokenization and encoding words

At the core of an LLM is a neural network, a computational system that transforms words into numbers for processing, creating a vast map of language where words and their meanings are stored as numerical patterns. This process begins with tokenization, which breaks text into manageable pieces, followed by encoding, which positions these pieces on the map using parameters to capture relationships, like how "bank" connects to "river" and "savings."

Tokenization: Breaking text into pieces

Tokenization splits text into smaller units called tokens, the building blocks the model analyzes. Think of tokens as puzzle pieces the model assembles to understand a sentence. A token can be a whole word (e.g., "bank," "river"), a part of a word (e.g., "un," "able" in "unable"), or punctuation (e.g., a comma). For example, in "The bank offers savings accounts," the model might tokenize it as:

- ["The", "bank", "offers", "savings", "accounts", "."]

In a contract's "liquidated damages" clause, like "The breaching party shall pay liquidated damages of $10,000," "liquidated damages" might be tokenized as a single token or split into ["liquidated", "damages"], depending on training. Tokenization is like chopping a legal document into bite-sized pieces, ensuring the model can process complex texts like contracts, statutes, or briefs.

In legal contexts, tokenization is critical. Specialized terms (e.g., "non-disclosure agreement," "UCC" for Uniform Commercial Code) or clauses (e.g., "force majeure," "liquidated damages") must be tokenized correctly. If "liquidated damages" is split improperly, the model might misinterpret it as unrelated terms, leading to errors in assessing breach penalties or contract enforceability.

Encoding tokens and their relationships

Once tokenized, each token is turned into a set of numbers through embedding, creating a numerical "address" on the language map. These numbers, stored as parameters, determine how tokens relate to each other. Unlike a 3D space with x, y, z axes, this map is a high-dimensional matrix with hundreds or thousands of axes – directions capturing subtle linguistic nuances. Think of it as a legal dictionary, where each token is an entry, and axes are definitions like "contract law," "tort liability," or "damages." This high-dimensionality makes LLMs "smart," enabling precise understanding of complex legal concepts.

For example, the token "bank" has multiple meanings: a river's edge ("sitting by the river bank") or a financial institution ("depositing money at the bank"). Its embedding might be [0.2, -0.5, 0.8, ...], positioning "bank" near:

- Nature-related tokens: "river," "creek," "stream," for geographic contexts.
- Finance-related tokens: "savings," "loan," "financial institution," for financial contexts.

Each axis might represent a feature, like "is this token related to water?" or "is it financial?" The thousands of axes allow fine distinctions, placing "bank" closer to "river" on some axes and "savings" on others, unlike "car." Similarly, "liquidated damages" might be encoded near "breach," "penalty," or "contract remedy," capturing its role in disputes. This nuanced encoding, learned from billions of sentences, enables LLMs to grasp relationships like "contract" being near "agreement," "clause," and "breach."

Copyright © 2025 by the author.
For reprint permission, contact the publisher: **www.plaintiffmagazine.com**



The attention mechanism: Using context to choose meanings

After tokens are encoded, the LLM determines which meaning of a token like "bank" fits the sentence. The attention mechanism acts like a spotlight, focusing on the most relevant tokens to interpret context, akin to a legal research team where researchers weigh different precedents.

How attention works

Attention weighs the importance of all tokens in a sentence for each token's meaning. In "She sat by the bank of the river," attention notices "river" and "of" suggest "bank" means a riverbank. Here's how:

- Token representations: Each token has its embedding (e.g., "bank" as [0.2, -0.5, 0.8, ...], "river" as [0.3, -0.4, 0.9, ...]).
- Comparing tokens: The model compares "bank" to other tokens, calculating relevance scores. "Bank" and "river" are close in the embedding matrix, so "river" gets a high score.
- Focusing the spotlight: Scores assign weights. "River" gets heavy weight, focusing attention on it. Less relevant tokens like "she" get lower weight.
- Updating the meaning: The model combines information, emphasizing "river," to shift "bank" toward riverbank.

In a legal context, consider a contract term: "The seller's failure to deliver constitutes a material breach." Attention focuses on "failure," "deliver," and "breach" to interpret "material breach" as a significant violation. However, if the contract includes conflicting clauses (e.g., "minor breach" elsewhere), attention might overemphasize irrelevant tokens, misinterpreting "material breach" as less severe. This occurs across multiple layers of the transformer architecture, where attention "heads" act like researchers analyzing intent, precedent, or liability, ensuring a comprehensive interpretation.

Legal implications

Attention enables LLMs to analyze complex legal texts, such as interpreting "force majeure" in "The bank shall indemnify the client for losses due to force majeure." However, ambiguous terms or lengthy documents can confuse attention. In the "material breach" example, misinterpretation could lead to incorrect liability assessments in a contract dispute. Incorrect tokenization (e.g., splitting "force majeure") exacerbates errors.

Training LLMs: Building the foundation

Training an LLM adjusts its parameters to predict language patterns, creating a foundation for novel responses. It involves pre-training and fine-tuning.

Pre-training

Pre-training teaches general language patterns using a massive dataset (e.g., books, websites, legal corpora). The goal is language modeling: predicting the next token. For "The court ruled that the contract was...", the model predicts "valid" if that's the actual next word.

Here's how:

- Data preparation: The dataset is tokenized and fed into the model.
- Forward pass: The model predicts the next token (e.g., guessing "breached").
- Loss calculation: The model checks its prediction against the dataset's actual token (e.g., "valid"), like a student comparing an answer to a textbook. The loss function, a mathematical formula, quantifies the error – how far "breached" is from "valid." A lower loss means a closer prediction. No human decides this; the dataset serves as the answer key, and the loss function automates the evaluation.
- Backpropagation: The model adjusts parameters to reduce loss, like the student revising study notes. It works backward, nudging parameters (e.g., "bank's" embedding numbers) using gradient descent, which follows the steepest path to lower loss.
- Iteration: This repeats over billions of tokens, requiring vast computational resources.

This automated process ensures the model learns patterns without human intervention, relying on the dataset's "ground truth" (e.g., "valid") to guide improvements.

The generative nature and copyright implications

The pre-training dataset – often including copyrighted material like novels, articles, or legal briefs – forms a rich, embedded knowledge base. LLMs don't store or reproduce this data verbatim. They tokenize and encode it into a high-dimensional embedding matrix, capturing relationships (e.g., "contract" with "breach," "hero" with "quest"). When responding to a prompt, the model generates novel outputs based on these patterns, not retrieved data, earning the term "generative."

This complicates copyright litigation. Consider a case where an LLM trains on a copyrighted poem. The poem is tokenized (e.g., ["moon," "June," "spoon"]) and embedded. If the LLM generates a new poem with similar patterns, is it infringement?

In *The Authors Guild, Inc. v. Google, Inc.* (2d Cir. 2015) 804 F.3d 202, the court found Google's book-scanning project "transformative" fair use because it provided searchable snippets without substituting for the original works. But LLM training differs significantly: While Google indexed books to help users find them, LLMs absorb patterns to generate new content. This distinction may matter – Google's service directed users to purchase books, while LLMs might produce content that competes with original works. Courts have not yet resolved whether encoding copyrighted material into parameters constitutes a derivative work or fair use, making this an evolving area of law.

Fine-tuning

Fine-tuning tailors the model for tasks like legal research, using smaller datasets (e.g., court opinions). It prioritizes legal patterns while retaining general knowledge.

Copyright © 2025 by the author.
For reprint permission, contact the publisher: **www.plaintiffmagazine.com**



Legal considerations

Training poses challenges:

- Data bias: If training data reflects societal biases, the model's outputs will likely reflect those same biases.
- Confidentiality: Fine-tuning on confidential documents risks data exposure.
- Transparency: Proprietary datasets obscure learning, complicating validation.
- Copyright and ethics: Training on copyrighted or privileged data raises legal and ethical risks.

Reinforcement learning: Aligning LLMs with human values

Pre-training and fine-tuning produce capable models, but outputs may not align with professional expectations – they might be too verbose, too casual, or miss the nuance lawyers need. Reinforcement learning (RL) refines behavior using human feedback to better match real-world requirements.

Reinforcement Learning from Human Feedback (RLHF)

RLHF improves model performance through an iterative process:

- Human feedback: Legal professionals evaluate multiple model outputs for the same prompt, ranking them based on accuracy, clarity, relevance, and professionalism.
- Reward model: A separate AI model learns to predict human preferences from these rankings, essentially learning what makes a "good" legal response.
- Policy optimization: The main LLM is fine-tuned using proximal policy optimization (PPO) to maximize these reward scores while avoiding drastic behavioral changes.

Example in legal context

Consider an LLM asked to summarize *Miranda v. Arizona* for a client memo. The model might generate three versions:

Version A (overly technical): "The Court held that the prosecution may not use statements stemming from custodial interrogation of the defendant unless it demonstrates the use of procedural safeguards effective to secure the privilege against self-incrimination, specifically requiring notification of the right to remain silent and the right to appointed counsel prior to interrogation."

Version B (too casual): "Cops have to tell you about your rights before they ask you questions when you're arrested."

Version C (balanced): "When police arrest and question suspects, they must first inform them of their constitutional rights: the right to remain silent, that anything said can be used against them in court, the right to an attorney, and the right to have an attorney appointed if they cannot afford one. These are known as Miranda warnings."

Through RLHF, human evaluators would likely rank Version C highest because it balances legal accuracy with client accessibility. The reward model learns these preferences across thousands of examples, teaching the LLM to produce responses that are both precise and comprehensible.

The process in practice

This isn't just about choosing between existing options. RLHF shapes how the model generates entirely new responses. For instance, after RLHF training on legal documents, an LLM learns to:

- Use appropriate legal terminology without overwhelming lay readers
- Include relevant citations while maintaining readability
- Structure arguments logically with clear topic sentences
- Avoid colloquialisms in formal documents while remaining clear

Legal implications

RLHF offers both benefits and challenges for legal practice:

Benefits:

- Consistency: Models learn to maintain professional tone across diverse legal tasks
- Practical alignment: Outputs better match what lawyers actually need
- Error reduction: Human feedback helps identify and reduce common mistakes

Risks:

- Hidden biases: If evaluators share similar backgrounds or preferences, the model may learn systemic biases (e.g., favoring certain legal writing styles or arguments)
- Accountability gaps: When errors occur, it's difficult to determine whether they stem from training data, the reward model's interpretation, or the optimization process
- Oversimplification: Models might learn to favor clarity over necessary complexity, potentially missing crucial legal nuances

For legal professionals, understanding RLHF is crucial because it explains both why LLMs can produce remarkably lawyer-like text and why they might consistently make certain types of errors. The human feedback that shapes these models comes with human limitations and biases built in.

Practical applications and ethical considerations

LLMs can transform legal practice through:

- Document review: Automating contract analysis.
- Legal research: Summarizing cases or identifying statutes.
- Drafting: Generating pleadings or memos.
- Client interaction: Powering chatbots for advice.
- E-discovery: Analyzing vast document sets for evidence.
- Litigation: Drafting responses to motions or briefs.

LLMs in e-discovery

In e-discovery, LLMs revolutionize document review by identifying relevant emails, contracts, or memos in litigation, going beyond traditional Boolean searches (e.g., keyword queries like "breach" or "discrimination"). Unlike Boolean methods, which rely on exact matches, LLMs understand context and semantic relationships, enabling deeper analysis of nuanced language patterns across documents. For example, in a workplace discrimination lawsuit, an LLM might flag

Copyright © 2025 by the author.
For reprint permission, contact the publisher: **www.plaintiffmagazine.com**



emails discussing "workplace bias" but also detect subtle patterns in performance evaluations that Boolean searches miss.

Consider a case where a company faces allegations of gender discrimination in promotions. An LLM can analyze thousands of performance evaluations to identify biased language: male employees might be described as "assertive," "bold," or "leadership material," while female employees are labeled "supportive," "reliable," or "team-oriented," suggesting they are relegated to background roles rather than challenged with leadership opportunities. By comparing linguistic patterns across documents, the LLM uncovers systemic bias that might not appear in individual evaluations or keyword searches, providing critical evidence for the plaintiff's case. This contextual understanding allows LLMs to reveal hidden issues in litigation or compliance, such as discriminatory practices or contract breaches embedded in subtle language.

However, LLMs' strengths come with risks. Biases in training data could skew relevance rankings. For instance, if training data underrepresents minority perspectives, the LLM might overlook evaluations describing minority employees' contributions, missing evidence of discrimination. Hallucination is another concern: An LLM might misinterpret a vague email as evidence of bias or fabricate a summary of a non-existent document. In one case, an LLM might incorrectly flag an email as "discriminatory" based on ambiguous phrasing, leading to false positives.

To mitigate these risks, lawyers must verify LLM outputs using e-discovery platforms like DISCO, Everlaw, or Relativity. These tools complement LLMs by providing structured review workflows, metadata analysis (e.g., email senders, dates), and manual oversight to confirm relevance and accuracy. For example, after an LLM flags performance evaluations for biased language, a lawyer can use DISCO to review the original documents, check metadata (e.g., reviewer's identity), and validate the LLM's findings against the case's legal standard. Everlaw's analytics features, like concept clustering, can further refine the LLM's contextual analysis, ensuring no relevant documents are missed. This verification process helps ensure that LLM outputs meet legal standards and avoid errors that could undermine a case.

A practical workflow for e-discovery with LLMs includes:

- Run LLM analysis: Use the LLM to identify relevant documents and detect patterns (e.g., biased language in evaluations).
- Export results: Transfer flagged documents to DISCO, Everlaw, or Relativity.
- Review metadata: Check document details (e.g., author, date) to confirm context.
- Validate relevance: Manually review LLM-flagged documents to ensure accuracy.
- Cross-check patterns: Use platform analytics to verify linguistic patterns (e.g., gendered descriptors).
- Document findings: Record the verification process for court scrutiny.

This workflow ensures LLMs' contextual insights are harnessed while mitigating risks, enhancing e-discovery's efficiency and depth.

LLMs in litigation: Responding to motions

In litigation, LLMs assist lawyers by drafting responses to motions, such as oppositions to summary judgment. Consider a California premises-liability case where a plaintiff trips on a sidewalk crack, suffers an injury, and sues the city. The city files a motion for summary judgment, arguing the .75-inch crack is a "trivial defect," which is not actionable under California's "trivial-defect doctrine." The plaintiff's lawyer uploads the motion and the text of *Caloroso v. Hathaway* (2004) 122 Cal.App.4th 922, a trivial-defect case, with the prompt: "Provide the best arguments to oppose this motion, including relevant California case law." Here's how the LLM processes the request:

- Parsing the motion: The LLM tokenizes the motion's text, breaking it into tokens like ["trivial," "defect," "sidewalk," "premises," "liability"]. Using embeddings, it maps these to a high-dimensional space, where "trivial defect" is near "minor defect" or "non-actionable." The attention mechanism focuses on key arguments: the city claims the crack's size is trivial, no aggravating factors exist, and cites *Caloroso* for precedent.
- Generating counterarguments: Drawing on training data (e.g., California premises-liability cases), the LLM suggests counterarguments:
- Aggravating factors: Argue that poor lighting, shadows, or debris made the crack dangerous, citing *Caloroso v. Hathaway* (2004) 122 Cal.App.4th 922, which held that triviality depends on all circumstances (e.g., visibility, traffic).
- Factual dispute: Highlight disputed evidence, such as witness testimony or photos showing the crack's deceptive appearance, to preclude summary judgment, per *Aguilar v. Atlantic Richfield Co*. (2001) 25 Cal.4th 826.
- Case law precedent: Reference *Fielder v. City of Glendale* (1977) 71 Cal.App.3d 719, where a 1-inch defect was non-trivial due to heavy foot traffic.
- Structuring a response: The LLM drafts a response brief, integrating these arguments and citations, formatted like a legal pleading.

However, LLMs' generative nature introduces significant risks, particularly hallucination of case citations and quotations, even with access to free databases like Justia or Google Scholar, or when case texts are provided. Hallucinations occur because LLMs predict tokens based on training patterns, not real-time verification, and their training data may include errors or fictional citations. For example, the LLM might cite a nonexistent "*Smith v. City of Oakland* (2023) 56 Cal.App.5th 789," mimicking citation formats from training data, or misquote *Caloroso* as stating, "even minor

Copyright © 2025 by the author.
For reprint permission, contact the publisher: **www.plaintiffmagazine.com**



defects are actionable if obscured by shadows," despite having the case text.

This happens because LLMs prioritize plausibility over fidelity to sources, relying on internalized patterns rather than analyzing provided texts directly. Even advanced models exhibit overconfidence, with some studies showing hallucination rates of 69-88% for legal queries, and may misinterpret context, conflating "defect" with unrelated legal doctrines.

Critically, LLMs can misattribute quotations even when they are "reading" the case it is discussing. In the *Caloroso* example, the LLM might fabricate a quote because it generates responses from patterns, not by cross-referencing the exact text, lacking the reasoning to prioritize source accuracy. When prompted to verify (e.g., "Is this quote accurate?"), LLMs can search databases or reanalyze provided texts, correcting errors reactively – or not! They may continue to hallucinate, even as they promise that they are citing the text accurately.

But they often don't verify proactively due to computational constraints, prompt-driven behavior, and no self-awareness to question outputs. This reactive verification explains why LLMs can (sometimes) recognize and correct mistakes *post hoc* but not preemptively, posing risks in legal practice.

Lawyers must verify all citations and quotes to ensure accuracy. In the well-publicized case of *Mata v. Avianca, Inc.* (S.D.N.Y. 2023) 678 F.Supp.3d 443, a lawyer's inclusion of hallucinated citations led to a $5,000 sanction. Today, unverified hallucinations can weaken arguments, or lead to dismissal and/or sanctions.
A verification workflow includes:

- Extract citations: List *all* LLM-provided case citations and quotes.
- Search databases: Use Westlaw or Lexis Nexis to confirm case existence and details.
- Validate quotes: Check pinpoint cites and quotes against original texts.
- Cross-check arguments: Ensure cited cases support the LLM's arguments.
- Document verification: Record the process for ethical accountability.

This process ensures LLMs' litigation support is reliable, preventing responses that could violate professional standards or harm case outcomes.

Ethical AI in legal practice

Responsible LLM use requires ethical frameworks, such as the ABA's Model Rules and emerging AI guidelines. Firms must establish policies for LLM deployment, ensuring outputs are verified, client data is protected, and biases are mitigated. For example, the ABA's 2019 Resolution 112 urges lawyers to address AI's ethical challenges. Training staff on LLM limitations, conducting regular audits, and using transparent tools enhance accountability.

Ethical issues include:

- Competence: Verifying outputs to avoid malpractice.
- Confidentiality: Inputting client data into cloud-based LLMs risks violating confidentiality. Know how the model you are using treats your data. Does it train on inputs?
- Bias and fairness: Biased outputs resulting from biases in training data may result in discriminatory outputs.
- Transparency: Courts demand AI explanations, but LLMs' opacity hinders this.
- Copyright: Training on copyrighted data raises risks for creative sectors.

Effective "prompt engineering" for legal tasks

How you phrase prompts – that is, the queries you make to the LLM – significantly affects LLM outputs. Consider these strategies:

- Be specific: Instead of "Summarize this contract," try "Identify all termination clauses in this contract and explain the conditions that trigger each one."
- Provide context: "Acting as a California employment lawyer, analyze whether this termination violates wrongful termination laws, considering at-will employment exceptions."
- Request structure: "List each argument in IRAC format" or "Provide your analysis in numbered paragraphs with case citations for each point."
- Iterative refinement: Start broad, then narrow: "What are the elements of breach of contract?" followed by "How does the economic-loss doctrine apply to negligent misrepresentation claims in construction defect cases?"
- Verification prompts: Always follow up with "List all case citations you provided so I can verify them" or "What specific page/section of the statute supports this interpretation." *Then, follow through with the verification!*

Conclusion

Large language models offer transformative potential for law, from contract analysis to e-discovery and litigation, but their complexity and risks demand scrutiny. Understanding tokenization, high-dimensional embeddings, attention, generative training, reinforcement learning, and limitations like hallucination and source unawareness empowers lawyers to leverage LLMs' benefits – efficiency, scalability, insight – while mitigating risks – bias, inaccuracy, and ethical concerns.


Ehrlich

Jeffrey I. Ehrlich is the principal of the Ehrlich Law Firm, APC, *in Claremont, California. He is certified by the State Bar of California as an Appellate Specialist. He is the editor-in-chief of Advocate Magazine, is an Emeritus Member of the CAALA Board of Governors, and is the Chair of the California Academy of Appellate Lawyers' Task Force on Generative AI.*

Copyright © 2025 by the author.
For reprint permission, contact the publisher: **www.plaintiffmagazine.com**