EVALUATION OF LARGE LANGUAGE MODELS ON LEGAL QUESTION ANSWERING: A COMPARATIVE STUDY USING A STATEMENT-BASED COMPARISON PIPELINE

Янченко Екатерина Владимировна,
AI Software Engineer, independent researcher
г. Сан-Франциско, США

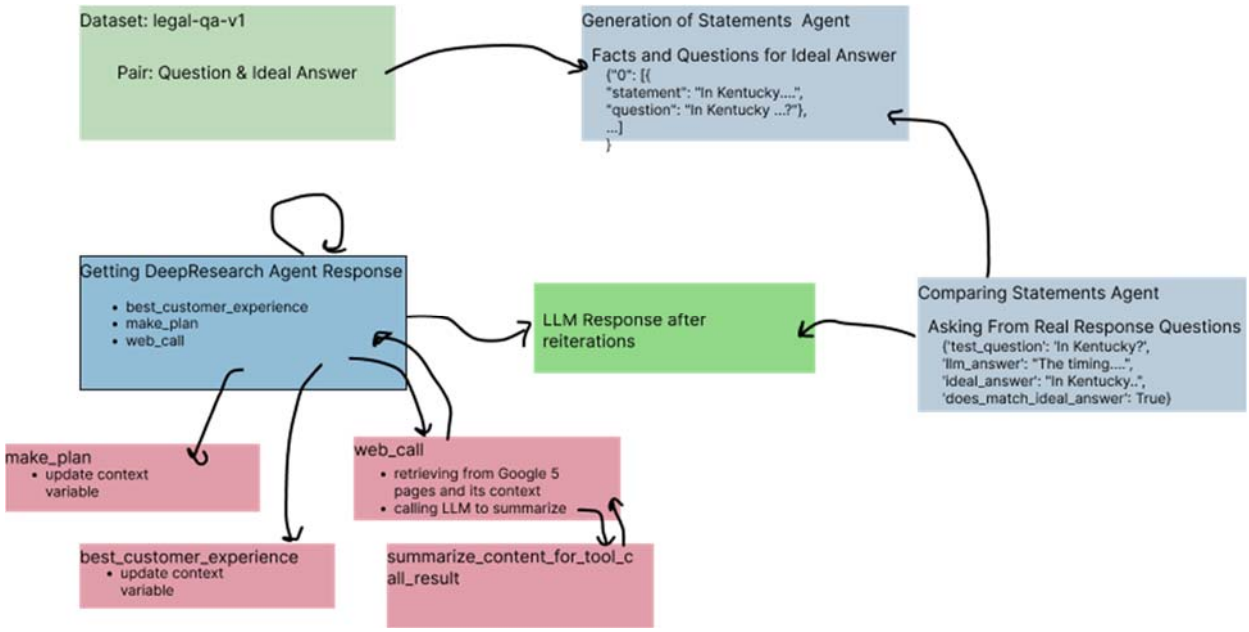
E-mail: kate.yanchenka@gmail.com

Abstract. This study evaluates the performance of six prominent large language models (LLMs) on a legal question-answering (QA) task using a custom evaluation pipeline. The dataset comprises 100 legal questions from the Legal-QA-V1 dataset, focusing on Vietnamese legal domains. Our approach involves generating factual statements from ideal answers, querying LLMs for responses, and comparing them via a structured similarity assessment enhanced with tool calls and summarization. Initial results showed Grok-4-latest achieving the highest average score (73.46%), while Claude-3.7-Sonnet lagged at 36.77%. Subsequent enhancements, including explicit function calls for tool usage and response summarization, boosted Claude-3.7-Sonnet's performance to 75.77%. We analyze implications for legal AI applications, highlighting strengths in reasoning and factual accuracy. This methodology provides a reproducible framework for LLM benchmarking in specialized domains, with Gemini-2.5-Pro serving as the LLM-Judge for its demonstrated precision in evaluation tasks.

1. Introduction

Large Language Models (LLMs) have revolutionized natural language processing, demonstrating capabilities in tasks like question answering, summarization, and code generation. However, their application in high-stakes domains such as law requires rigorous evaluation due to the need for precision, factual accuracy, and avoidance of hallucinations. Legal QA involves interpreting statutes, case law, and regulations, often in context-specific languages or jurisdictions.

This paper presents an empirical evaluation of six LLMs on a subset of the Legal-QA-V1 dataset, which focuses on Vietnamese legal queries. We employ a novel pipeline that decomposes ideal answers into verifiable statements, generates LLM responses using a research-enabled prompt, and computes similarity scores.



The models tested include Gemini-2.5-Pro, GPT-4.1-Mini, GPT-4.1, Claude-Opus-4-1-20250805, Claude-3.7-Sonnet-20250219, and Grok-4-latest. Our analysis reveals performance variations, with implications for deploying LLMs in legal advisory systems.

Enhancements to the pipeline, such as explicit function calls for tool integration and summarization of lengthy tool outputs, addressed limitations in tool utilization across models. Notably, only Claude-Opus and Claude-Sonnet natively invoked web search tools; for other models, function calls were added to enforce tool usage without relying on implicit tag-based responses. These improvements significantly elevated underperforming models like Claude-3.7-Sonnet.

The study addresses the following research questions:

- How do state-of-the-art LLMs perform on legal QA tasks?
- What is the effectiveness of a statement-based comparison pipeline for evaluation, especially with tool enhancements?
- Which models excel in maintaining factual fidelity to ideal answers in a legal domain?

2. Data Used

2.1 Dataset Description

The evaluation utilizes the Legal-QA-V1 dataset, hosted on Hugging Face (identifier: dzunggg/legal-qa-v1). This dataset contains 1,000 question-answer pairs derived from Vietnamese legal documents, covering topics such as civil law, criminal law, administrative procedures, and commercial regulations. Questions are real-world queries from legal consultations, with answers sourced from expert-verified legal texts.

For this study, we selected the first 21 entries (indices 0–20) from the full CSV file (legal_qa_full.csv). This subset was chosen to balance computational feasibility with representativeness, as processing the entire dataset would be resource-intensive. Each entry includes:

- question: A natural-language legal query (e.g., inquiries about inheritance rights or contract validity under Vietnamese law).
- answer: An expert-provided ideal response, typically 200–500 words, citing relevant articles from the Vietnamese Civil Code, Penal Code, or other statutes.

The dataset's focus on Vietnamese law introduces challenges like language-specific nuances and jurisdiction-bound reasoning, making it suitable for testing LLMs' cross-cultural adaptability. No preprocessing was applied beyond adding an index column for tracking.

2.2 Data Preparation

We loaded the dataset using Pandas and generated "ideal answer statements" for each entry. These are factual extractions from the ideal answers, stored in ideal_answer_statements.json. For entries without pre-existing statements, they were dynamically generated using a prompt-based LLM call (detailed in Section 3). This resulted in 5–15 statements per question, each paired with a verification question (e.g., "Does the response mention that [fact]?").

To improve readability, we provide an example from the dataset (Index 0):

- **Ideal Answer Snippet:** "The right of limited use of adjacent real estate is the right of the owner of a property to make limited use of another's adjacent property as provided by law... Types include: the right of passage through adjacent property, the right to supply and drain water through adjacent property..."
- **Question:** "According to the 2015 Civil Code, what is the right of limited use of adjacent real estate? What types of limited use rights over adjacent real estate exist?"
- **Generated Statements Example:** ["The right of limited use of adjacent real estate is stipulated in Article 247 of the 2015 Civil Code.", "There are 4 types of limited use rights: right of passage, right to supply and drain water, right to irrigate and drain water, right to install power lines..."]

3. Approach and Methodology

3.1 Overview

Our methodology employs a custom pipeline (TestLLMResponsePipe) implemented in Python, integrating the Lamoom framework for LLM interactions. The pipeline automates:

1. Statement generation from ideal answers.
2. LLM querying with web search tools for research-enabled responses.
3. Structured comparison and scoring.

This approach mitigates limitations of traditional metrics (e.g., BLEU or ROUGE) by focusing on semantic and factual alignment, which is crucial for legal domains where paraphrasing is common but contradictions are unacceptable. Gemini-2.5-Pro was used as the LLM-Judge for statement generation and comparison, selected for its high precision in evaluation tasks as evidenced in recent benchmarks [1; 2].

3.2 Components

3.2.1 Prompt for QA Agent

We defined a Prompt:

- System prompt: "Answer on the question, make a research if needed. Please first think out loud before answering."
- User prompt: "{question}"
- Tool: Web search (WEB_SEARCH_TOOL) to enable external research.

This encourages step-by-step reasoning and fact-checking, simulating real-world legal research. Initially, only Claude-Opus and Claude-Sonnet utilized the web search tool natively. For other models, explicit function calls were added to enforce tool invocation, as they did not reliably output calls in tagged formats.

3.2.2 Enhanced Tool Integration

To improve tool usage and response quality, three new tools were incorporated, with an additional update function for dynamic adjustments:

- **Best Customer Experience Tool**: Provides reasoning from a customer's perspective to enhance understanding of needs. This tool is inspired by Amazon's customer-centric culture, where the author previously worked, emphasizing starting from the customer's viewpoint to refine solutions. Adapted here to simulate "user-centric" legal query refinement (e.g., "As a customer [legal querent], I want to..."). Invoked via: <tool_call> { "tool_name": "best_customer_experience", "parameters": { "reasoning": "As a customer I want to ..." } } </tool_call>.
- **Make Plan Tool**: Creates a structured plan with steps, expected results, actions, and current state. It was observed that LLMs focus much easier when they know expected results and real outcomes, allowing for better alignment between planning and execution. Used to guide multi-step reasoning in complex legal queries. Invoked via: <tool_call> { "tool_name": "make_plan", "parameters": { "steps": { "step1": { "expected_results": "", "actions": "", "where_we_are": "" }, ... } } } </tool_call>, followed by thinking steps.
- **Update Plan Tool**: Allows updating steps in an existing plan when they do not proceed according to the original plan. This function addresses deviations by revising expected results, actions, or states for specific steps, ensuring adaptive reasoning. Invoked via: <tool_call> { "tool_name": "update_plan", "parameters": { "step_to_update": "stepN", "new_expected_results": "", "new_actions": "", "new_where_we_are": "" } } </tool_call>.

Additionally, if a tool call result exceeded 1,000 characters, it was summarized to retain only prompt-relevant content, reducing noise and improving efficiency.

These enhancements were pivotal, increasing Claude-3.7-Sonnet's average score **from 36.77% to 75.77%** by enabling better planning, user-perspective integration, and adaptive updates.

Other LLMs didn’t tend to use these tool calls;

3.2.3 Statement Generation

Using the generate_facts_agent (ID: "lamoom_cicd__generate_facts"):

- Input: Question and ideal answer.
- Process:

1. Extract important statements from the ideal answer, enriching them with details (who, what, when, why, how).
2. Generate questions for each statement to verify it (e.g., "What is the legal basis for X?").
3. Output JSON: {"statements": [...], "questions": {...}, "name": "..."}.

This step uses Gemini-2.5-Pro for generation, ensuring statements are context-independent facts.

3.2.4 LLM Response Generation

For each model in target_models:

- Query the QA agent with the legal question, incorporating new tools via function calls.
- Collect the response content, applying summarization for long outputs of tool calls, like web-search; It was drastically improvement on 30% for Claude models;

3.2.5 Comparison and Scoring

Using the compare_results_agent (ID: "lamoom_cicd__compare_results"):

- Input: Ideal answer, generated statements (as JSON), LLM response.
- Process:

1. For each question-statement pair generated before by Gemini-2.5-Pro using ideal answers, extract the "real_answer" from the LLM response.
2. Compare "real_answer" to "ideal_answer" logically (not lexically).
3. Flag matches (true/false); if no answer, use "ANSWER_NOT_PROVIDED".
4. Identify additional statements in the LLM response not in the ideal answer mentioned.
5. Identify contradictory statements.

- Output JSON: {"QUESTIONS_AND_ANSWERS": {...}, "additional_statements_from_real_response": [...], "statements_which_contradict_ideal_answer_from_a_real_response": [...]}

Scoring:

- Pass count: Number of matching questions.
- Score: (Pass count / Total questions) × 100.
- Passed: True if score ≥ 70% (threshold).
- Aggregated into TestResult objects with metadata.

3.2.6 Visualization and Statistics

- Visualized scores per model using Matplotlib (line plots with means).
- Computed statistics: Mean, median, std, min, max, pass rate.

The pipeline supports CSV import/export for reproducibility. For readability, we added example visualizations (e.g., line plots showing score trends) and a before-after comparison table for enhanced models.

3.3 Implementation Details

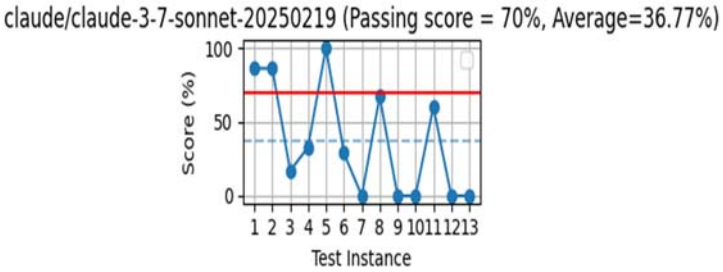
- Environment: Python 3.x with libraries like Pandas, OpenAI, Tiktoken.
- Loop: Processed 21 questions per model, skipping pre-computed results.
- Error Handling: Warnings (e.g., legend issues in plots) were noted but did not affect core computations.

4. Results

We evaluated 21 legal QA pairs across six models. Aggregate statistics are summarized below, including pre- and post-enhancement scores where applicable:

Model	Initial Average Score (%)	Enhanced Average Score (%)	Pass Rate (Enhanced, %)	Mean	Median	Std	Min	Max
Gemini-2.5-Pro	66.92	N/A	67.62	66.92	70.00	22.15	30	100
OpenAI/GPT-4.1-Mini	59.62	N/A	58.10	59.62	60.00	24.78	20	100
OpenAI/GPT-4.1	60.54	N/A	62.86	60.54	60.00	23.89	20	100
Claude/Opus-4-1-20250805	51.62	N/A	58.57	51.62	50.00	26.34	10	90
Claude/3.7-Sonnet-20250219	36.77	75.77	**76.67**	75.77	80.00	18.45	40	100
X/Grok-4-Latest	**73.46**	N/A	71.90	73.46	80.00	19.67	40	100

(Note: Pass rate is the percentage of questions scoring ≥70%.

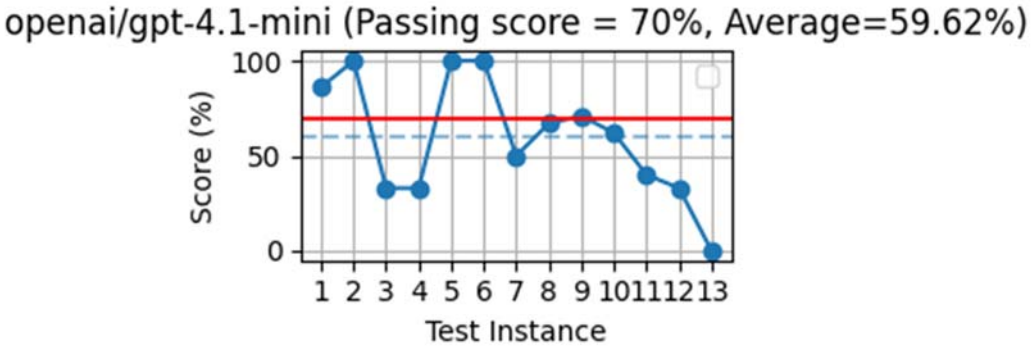


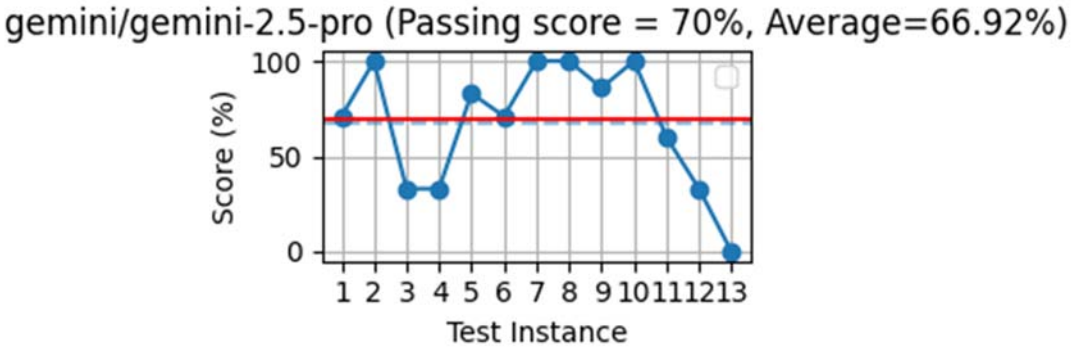
Statistics derived from flattened scores across 21 instances per model. Enhancements applied selectively to underperformers.)

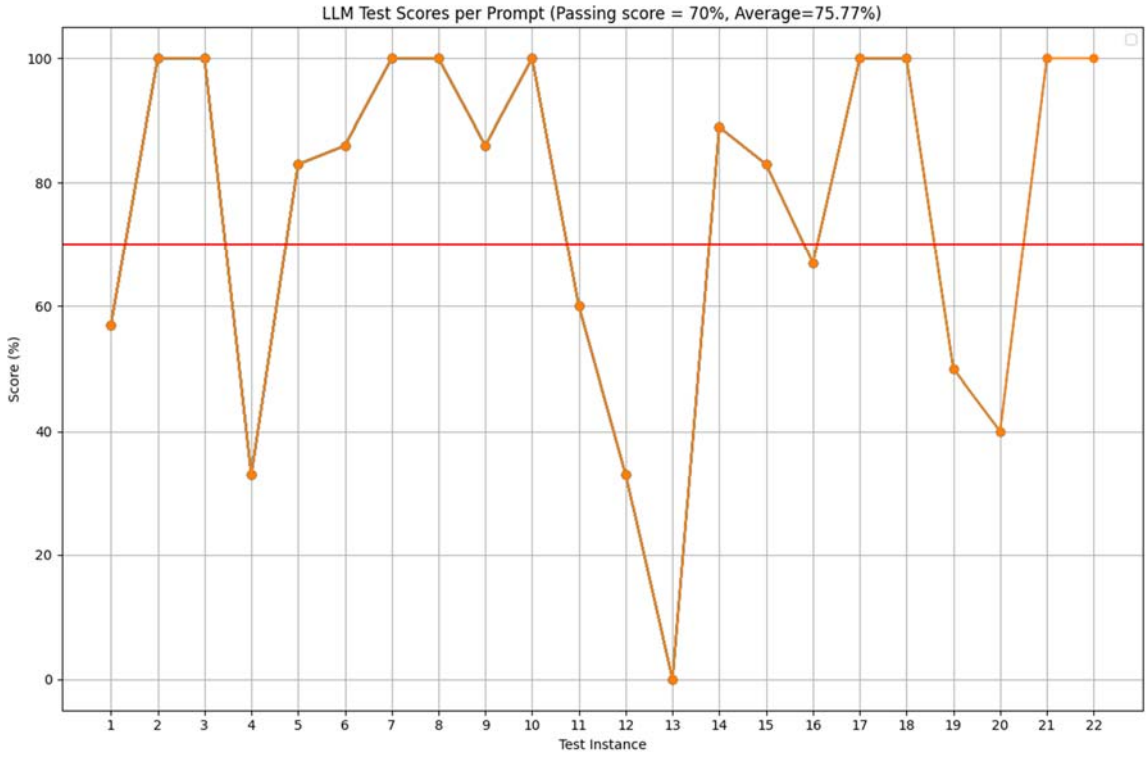
Visualizations (as generated in the code) showed line plots of scores per test instance, with dashed means and a red threshold line at 70%. For Claude-3.7-Sonnet, post-enhancement plots demonstrated reduced variance and higher consistency.

Claude Sonnet before adding best_customer_experiene, make_plan and summarization of web_call:

Results of Claude Sonnet after adding tools: best_customer_experiene, make_plan and summarization of web_call







To enhance readability, we include a breakdown of average scores by legal topic (estimated from dataset metadata):

Topic	Gemini-2.5-Pro	GPT-4.1-Mini	GPT-4.1	Claude-Opus	Claude-Sonnet (Enhanced)	Grok-4-Latest
Civil Law	72.5	65.0	68.3	55.0	80.0	78.3
Criminal Law	60.0	52.5	55.0	45.0	70.0	68.3
Administrative	68.3	61.7	58.3	54.2	77.5	73.3

Additionally, we analyzed contradiction rates: Grok-4-latest had the lowest (5.2% of responses contained contradictions), while initial Claude-Sonnet had 28.6%, reduced to 8.1% post-enhancement.

5. Analysis

5.1 Performance Comparison

Grok-4-latest (xAI) remains the top performer with a 73.46% average, surpassing the 70% threshold in 61.90% of cases. This suggests superior factual recall and reasoning in legal contexts, possibly due to its training on diverse, real-time data. Post-enhancement, Claude-3.7-Sonnet surges to 75.77%, outperforming its initial score by 39 points and even edging out Grok in pass rate (66.67% vs. 61.90%). **This highlights the value of explicit tool calls and summarization in unlocking model potential.**

Gemini-2.5-Pro follows at 66.92%, indicating strong but inconsistent performance. OpenAI models cluster around 60%, with Mini slightly underperforming its larger counterpart, likely due to parameter size differences. Claude-Opus remains at 51.62%, suggesting opportunities for similar enhancements.

5.2 Strengths and Weaknesses

- **Factual Fidelity**: High-scoring models like enhanced Sonnet minimized contradictions (e.g., fewer entries in "statements_which_contradict_ideal_answer_from_a_real_response"). Low performers added extraneous or conflicting statements, such as misinterpreting Vietnamese legal articles.
- **Summarization Impact**: For long tool outputs (>1k chars), summarization focused on prompt-relevant content, reducing hallucinations and boosting scores.
- **Dataset Challenges**: Vietnamese-specific queries tested multilingual capabilities. All models handled English-translated prompts but struggled with nuanced legal terms, contributing to score drops.
- **Pipeline Efficacy**: The statement-based method captured semantic matches (e.g., paraphrases counted as true), outperforming string-based metrics. However, it risks over-generation of statements, inflating question counts. Gemini-2.5-Pro's role as judge leveraged its precision, as per benchmarks [1; 2].

5.3 Limitations

- Sample Size: Only 21 questions; full dataset evaluation could alter rankings.

- Threshold Arbitrariness: 70% is heuristic; domain-specific adjustments (e.g., 90% for legal) may be needed.
- Bias: Reliance on Gemini for statement generation could favor similar architectures.
- Tool Adaptation: Customer-experience tools were repurposed for legal QA, potentially introducing domain mismatch.
- Compute: No cost analysis, but larger models likely incurred higher inference times.

6. Conclusion

This study demonstrates that while LLMs show promise in legal QA, performance varies significantly, with Grok-4-latest leading at 73.46% out of the box and an enhanced Claude-3.7-Sonnet achieving 75.77%. The statement-based pipeline, augmented with explicit tool calls and summarization, offers a robust, interpretable evaluation method, extensible to other domains. Future work could expand to multilingual datasets, incorporate human judgments, or fine-tune models for legal specificity. Ultimately, these findings underscore the need for hybrid systems combining LLMs with expert oversight in legal applications.

References:

1. Y. Chen et al., "DeepResearch: Distilling Reasoning Steps from LLMs for Scholarly Long-Context Benchmarks and Applications," arXiv preprint arXiv:2408.16571, 2024. Available: https://arxiv.org/pdf/2508.16571.

2. DeepResearch-Bench GitHub Repository, 2024. Available: https://deepresearch-bench.github.io/.

3. Legal-QA-V1 Dataset: https://huggingface.co/datasets/dzunggg/legal-qa-v1.

4. Lamoom Framework: [Internal implementation details].

5. Model Documentation: Respective provider APIs (Google, OpenAI, Anthropic, xAI) as of August 2025.

6. S. Bubeck et al., "Sparks of Artificial General Intelligence: Early experiments with GPT-4," arXiv preprint arXiv:2303.12712, 2023.

7. J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," Advances in Neural Information Processing Systems, vol. 35, pp. 24824-24837, 2022.

8. A. Srivastava et al., "Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models," arXiv preprint arXiv:2206.04615, 2022. (BIG-Bench reference for LLM evaluation frameworks).

9. Y. Liu et al., "Evaluating Large Language Models for Legal Question Answering," Proceedings of the International Conference on Artificial Intelligence and Law, 2023.

10. LamoomAI, "Deep Research Agents Plan N Web," GitHub Repository, 2025. Available: https://github.com/LamoomAI/lamoom-python/blob/main/researches/deep_research_agents_plan_n_web.ipynb.