



Hacking Legal Munich Challenge

26 – 27 April 2025

Task

Develop a Contract Analysis Tool to compare legal contracts (1x template, 1x new draft) for key clauses, potential conflicts, and risks.

The tool shall highlight areas that may require further legal review.



Requirements and Challenges

Requirements

1. **Language:** We recommend Python or JavaScript/TypeScript.
2. **Functionality:**
 - a. **Clause Extraction:** Extract and categorize key clauses (e.g., termination, liability, confidentiality) from a given legal contract.
 - b. **Conflict/Deviation Detection:** Highlight deviations/conflicts that may require clarification.
3. **User Interface:**
 - a. Build a simple web interface where users can upload DOCX and PDF files (and potentially paste contract text) and receive analyzed results.
 - b. Display results clearly, indicating deviations/conflicts and, where possible, additional explanatory information.
4. **Output:**
 - a. Generate a summary report for each uploaded contract, detailing:
 - i. Extracted clauses with categories.
 - ii. List of deviations/conflicts with suggestions for clarification.
5. **Documentation:**
 - a. Provide clear documentation on how to set up and run the tool.
 - b. Include a user guide explaining the interpretation of analysis results.

Challenges

- **Legal Understanding:** Code must attempt to mimic a lawyer's intuition in identifying significant legal aspects.
- **Clarity:** Output must be easily comprehensible for a non-technical audience
- **Error Handling:** Implement robust error handling.
- **Document Handling:** Ideally it will handle both DOCX and PDF inputs.

Navigating Legal Documents

- **Provided Legal Documents:** Two contracts (and for each, two versions) are provided for analysis. One version of each contract is the **standard version**, which serves as the baseline for comparison. The other version is labelled as **amended version**, representing a new draft with (suggested) changes. This models real-world use cases where legal professionals or parties to a contract receive and review contracts provided by one party with suggested changes, or where a standard version is updated internally.
- **Defined Terms:** Contracts often include "Defined Terms," which are specific words or phrases given a particular defined meaning within the document and that are highlighted in bold and capitalized throughout the entire contract (such as: *This Non-Disclosure and Confidentiality Agreement (the "**Agreement**")*)).
- **Disclaimer:** The provided contracts and documents were designed to provide data, assistance, and illustrative examples. All companies, persons, addresses, and other personal information included in the documents provided for this challenge are entirely fictional. None of the contracts have ever been entered into and do not depict any actual entities or individuals.



Deliverables and Evaluation Criteria

Deliverables

1. Source code with detailed inline comments.
2. Readme file with setup instructions, dependencies, and how to use the tool.
3. Example output for validation.
4. A brief report on challenges faced and how they were overcome.

Evaluation Criteria

- **Correctness & Coverage:** Accuracy in identifying and categorizing clauses, and deviations/conflicts.
- **Code Quality:** Clarity and organization of the code.
- **Usability:** Ease of use and intuitiveness of the interface and output.
- **Innovation:** Creative use of new (e.g. prompt engineering) and old (rule-based programming) approaches

An abstract graphic design featuring two overlapping triangular shapes in shades of blue. The lighter blue triangle is positioned in the upper left, while the darker blue triangle is in the lower left, creating a dynamic geometric composition. The right side of the image is a plain white background.

ELTE.MATE.COM