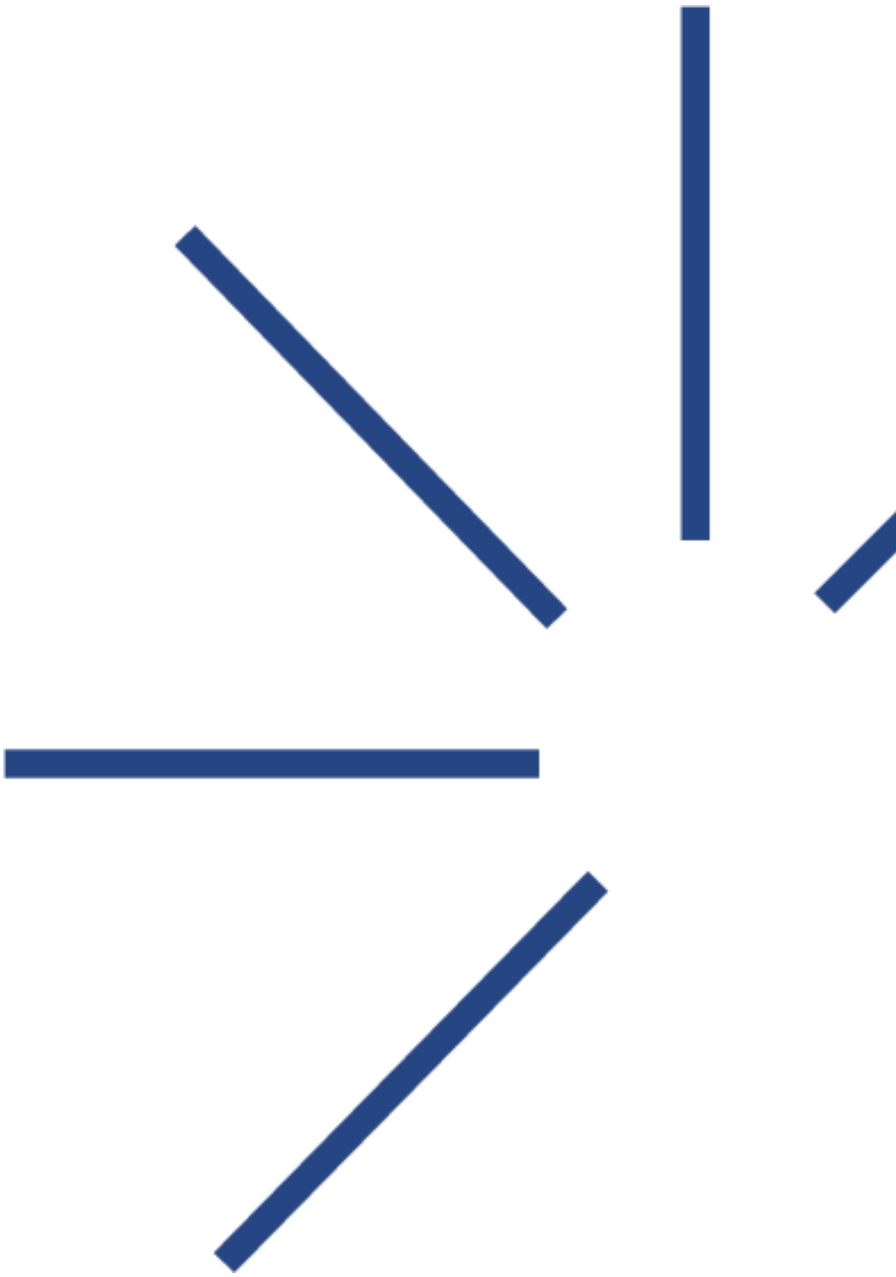
D06 – Method for verdict extraction (EN)



Project funded by the European Union's Next Generation EU through the Slovak Republic's Recovery and Resilience Plan under project no. 09I05-03-V02-00049.


Financované
Európskou úniou
NextGenerationEU

PLÁN OBNOVY

ÚRAD PODPREDSEDU VLÁDY
SLOVENSKEJ REPUBLIKY
PRE PLÁN OBNOVY
A ZNALOSTNÚ EKONOMIKU

VAIA
VÝSKUMNÁ
A INOVAČNÁ
AUTORITA

VÝSKUMNÁ
AGENTÚRA

`

INTRODUCTION

The project **"Automation of Legal Text Analysis based on Machine Learning"** (hereinafter "ALTAML") represents a key initiative aimed at integrating innovative approaches in the field of data processing and subsequent analysis, with a focus on the field of law, which also includes information and communication technology law. The aim of this project is to develop and validate effective methods for the automated analysis of legal texts using machine learning techniques. This involves, in particular, the development of tools that facilitate the processing and analysis of large volumes of data in the form of various legal documents, as well as the extraction of relevant information (attributes) from such documents, including the identification of key terms, references to legislation and other attributes.

The aim of the ALTAML project is therefore to contribute to more efficient access to legal information and to the acceleration of legal processes, thereby ensuring a higher degree of legal certainty and improving the availability of legal texts, the results of their analysis and relevant legal information for both professionals and the general public.

As part of the KPB2 work package, a deliverable entitled 'Method for verdict extraction' was produced. The proposed system is a hybrid data extractor for criminal law decisions, combining rule-based text processing, heuristic natural language processing methods, and a large language model deployed locally via the Ollama platform. The system is designed for the automated analysis of Slovak criminal decisions and the identification of structured legal data, including the statement of facts, sentences imposed, mitigating and aggravating circumstances, means of evidence, and the identification of relevant criminal offences.

The system features a multi-level architecture that combines deterministic rules, statistical text pre-processing and a large language model (LLM). The extraction of criminal offences is limited to a predefined database of Slovak criminal offences, thereby ensuring terminological consistency and minimising the risk of generating non-existent or legally incorrect labels. A brief description of the system is provided below.

Hybrid extractor of procedural data and criminal offences from court decisions

The extractor is a software module written in Python, designed to automatically structure data from Slovak criminal court decisions. The system uses a combination of rule-based methods, heuristic text analysis and the large language model LLaMA 3.3 to accurately extract legally relevant information.

System functionality

`

The main functionality of the system consists of the following steps:

1. Preparation of input data

The system works with a corpus of criminal court decisions in JSON format. The decisions contain the full text of the document, including the facts of the case, the legal assessment, the evidence and the court's ruling.

The ijson streaming parser is used to process large datasets efficiently; it allows decisions to be read sequentially without the need to load the entire dataset into main memory.

The decision texts are normalised through a simple cleaning process, which involves:

- removal of redundant spaces,
- normalisation of punctuation,
- removal of special characters,
- transforming the text into a uniform format.

2. Preliminary analysis of the decision

Before the extraction process begins, the system checks the number of defendants or offenders mentioned in the decision. This task is carried out using a language model that analyses the introductory section of the decision and determines whether the proceedings concern one or more offenders.

Decisions involving multiple offenders are automatically excluded from processing, thereby reducing the risk of incorrect attribution of data to individual persons.

3. Extraction of procedural data

The main data extraction is carried out using a large language model controlled by a strictly defined prompt formulated in the Slovak language.

The system extracts in particular:

- the statement of facts in the decision,
- information regarding acquittal,
- the penalties imposed,
- mitigating circumstances (Section 36 of the Criminal Code),

`

- aggravating circumstances (Section 37 of the Criminal Code),
- the evidence cited in the decision.

The statement of facts is extracted as faithfully as possible directly from the text of the decision without summarisation or rephrasing.

4. Extraction of criminal offences

The identification of criminal offences takes place in a separate, multi-stage processing phase.

4.1. Criminal offence database

The system uses an external JSON file containing a database of Slovak criminal offences.

Each record contains:

- the name of the criminal offence,
- the relevant section,
- a textual description of the facts of the offence.

This database represents a closed set of candidate offences from which the system can select.

4.2. Heuristic selection of candidates

Before sending the text to the language model, the system generates a set of candidate criminal offences based on lexical overlap between the factual sentence and the descriptions of criminal offences.

The following are used in calculating candidates:

- text tokenisation,
- removal of Slovak stop words,
- text normalisation,
- heuristic prioritisation of selected legal indicators.

The system includes specialised heuristics for frequently occurring factual situations, for example:

- failure to pay maintenance,
- making dangerous threats,
- damage to another person's property,

`

- acts committed under the influence of alcohol,
- fraudulent conduct.

The result of this phase is a limited number of the most relevant candidates, which proceed to the final decision-making stage of the language model.

5. Reranking using a language model

The final identification of criminal offences is carried out using the locally deployed LLaMA 3.3 language model via the Ollama platform.

The model is guided by a strictly defined prompt, which ensures:

- selection exclusively from the specified candidates,
- a ban on generating new names for criminal offences,
- the ability to identify multiple concurrent criminal offences,
- deterministic output in the form of a JSON structure.

In the event of a language model failure or an empty result, the system automatically activates a rule-based fallback mechanism based on heuristic rules and keywords.

6. Extraction of evidence and circumstances

The system identifies and categorises the evidence cited in the decision into several groups:

- interviews,
- documentary evidence,
- expert reports,
- physical evidence and records,
- other evidence.

Mitigating and aggravating circumstances are mapped to their full legal wording in accordance with the provisions of the Criminal Code. When a reference to a specific section of a legal provision is identified, the system automatically inserts the full text of the circumstance.

7. System output

The system output is a JSON structure containing:

- the decision identifier,
- information on whether the decision has been skipped,

`

- a statement of facts,
- identified criminal offences,
- the penalties imposed,
- mitigating and aggravating circumstances,
- categorised evidence.

Results are saved continuously during processing, thereby minimising the risk of data loss when processing large corpora.

8. Requirements and technologies

The system is implemented in Python and utilises the following technologies:

- Ollama for running the language model locally,
- the LLaMA 3.3 model for legal data extraction,
- JSON for streaming processing of large JSON datasets,
- requests for communication with the LLM API,
- regular expressions and heuristic rules for text pre-processing.

A minimum of 32 GB RAM is recommended. GPU (CUDA) support significantly speeds up language model inference when processing large legal corpora.

9. Source files

The project's source code and data file are maintained in a repository and are available at: https://gitlab.science.upjs.sk/altaml/verdict-extraction