D10 – Semi-automated method for the segmentation of legal texts (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 AKCELERAČNÁ 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 work package KPB2, the output 'Semi-automated method for the segmentation of legal texts' was developed. The proposed system represents a hybrid segmentation and classification model for legal texts, combining statistical natural language processing methods, semantic vector representations, heuristic rules and sequential post-processing. The aim of the system is to automatically divide Slovak court decisions into logical content sections and assign a legally relevant segment type to each section.

The system operates on multiple levels. First, it performs a coarse segmentation of the document into thematic blocks by analysing semantic changes between sentences. It then applies a hybrid classification mechanism based on a combination of TF-IDF similarity, embedding-based semantic similarity and rule-based heuristics. In the next phase, fine-grained sentence relabelling is carried out, which allows the boundaries of individual segments to be refined and local inconsistencies in classification to be removed. The result is a structured division of the decision into legally meaningful units representing the individual stages of judicial decision-making. A brief description of the system is provided below.

A hybrid system for the segmentation and classification of Slovak court decisions

The system is implemented in Python and is designed for the automatic processing of large corpora of Slovak court decisions. It combines methods of statistical text processing, transformer language models and heuristic rules created on the basis of the structure of Slovak legal documents.

System functionality

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

1. Preparation of input data

The system works with the texts of Slovak court decisions stored in a structured JSON format. The texts are first normalised and cleaned of technical artefacts, redundant spaces and special markers. The documents are then split into individual sentences using a rule-based segmenter tailored to legal texts.

1.1. Semantic representation of sentences

Each sentence in the document is transformed into a vector representation (embedding) using the SlovakBERT transformer language model, fine-tuned for the task of semantic similarity between sentences. The embeddings serve as the basis for identifying thematic changes within the document and the subsequent segmentation of the text.

1.2. Detection of segment boundaries

An analysis of semantic distances between neighbouring sentences is used to identify the boundaries between the content sections of the decision. The system calculates a thematic change score between sentence windows and determines candidate segment boundaries based on local maxima. The segmentation is further extended by macro-boundaries identified using structural markers typical of Slovak court decisions.

2. Segment classification

After creating the initial segments, the system classifies them into predefined legal categories:

- Subject_of_the_dispute
- Judgment_section
- Court_assessment
- Facts of the case
- Legal_framework
- Evidence
- Previous proceedings in the case
- Plaintiff_Prosecutor
- Defendant
- Other

2.1. TF-IDF representation of segments

Each segment is represented using character-based TF-IDF n-grams. Unlike traditional approaches, a character-based representation of the char_wb type is used, which allows for a more robust capture of morphological variants in Slovak legal language and reduces sensitivity to minor textual differences.

2.2. Semantic scoring

For each segment, similarity is calculated against the reference descriptions of individual classes. Reference descriptions represent the typical content of a specific legal category and serve as prototypical representations of classes.

2.3. Rule-based heuristics

The resulting segment scores are adjusted using an extensive set of heuristic rules based on:

- the occurrence of legal terms,
- procedural terms,
- references to legislation,
- identification of parties to proceedings,
- detection of operative provisions,
- analysis of judicial reasoning,
- the structural position of a segment within a document.

The rules enable classification accuracy to be improved in cases where semantic similarity alone does not provide a sufficiently reliable distinction between legal categories.

3. Sequential post-processing

Following the initial classification of segments, the system performs sequential post-processing, the aim of which is to remove inconsistencies between adjacent segments. Post-processing consists mainly of:

- corrections to incorrectly identified clauses,
- adjustments to segments based on their position in the document,
- merging short anomalous segments between blocks with similar content,
- corrections to transitional segments between individual legal sections.

This step allows us to take into account the sequential nature of legal documents and stabilise the resulting segmentation structure.

4. Sentence relabelling

To increase the accuracy of segment boundaries, a second level of classification is implemented at the sentence level.
Each sentence inherits the initial label of its parent segment and subsequently undergoes a separate relabelling process based on:

- embedding similarity,
- the local context of neighbouring sentences,
- sequential rules of the ABA and ABC type,
- heuristic rules for legal formulations.

The system iteratively corrects locally inconsistent sentence sequences and subsequently rejoins neighbouring sentences with the same labels into final segments.

5. System output

The system's output is a structured JSON representation of the document containing:

- segment identifier,
- segment type,
- the range of sentences in the segment,
- the number of sentences in the segment,
- segment text,
- classification score,
- alternative candidate labels.

The system also generates analytical CSV outputs containing relabelling statistics, transition matrices between classes, and aggregated segmentation metrics.

6. Requirements and technologies

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

- Sentence Transformers for sentence embeddings,
- SlovakBERT for the semantic representation of legal texts,
- scikit-learn for TF-IDF representations and cosine similarity,
- PyTorch for neural models and GPU acceleration,
- NumPy and Pandas for data processing and analysis,
- heuristic regular expressions for legal rules.

32 GB of RAM or more is recommended. GPU acceleration via CUDA significantly speeds up the calculation of embeddings and the processing of large legal corpora.

7. Source files

The project's source code and dataset are maintained in a repository and are available at: https://gitlab.science.upjs.sk/altaml/legal-text-segmentation