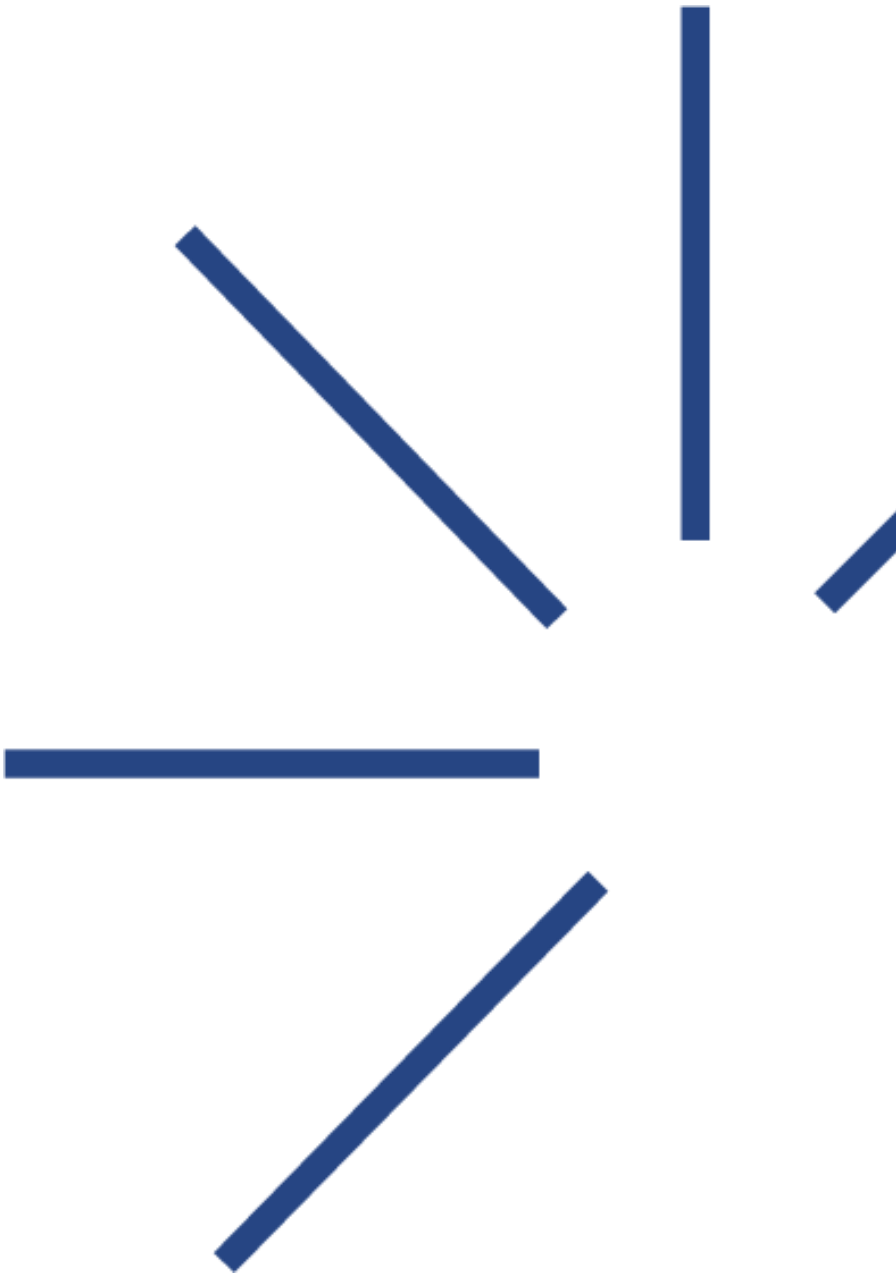
D19 – Algorithm for legal text linking (EN)



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
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 a more efficient approach to legal information and to the acceleration of legal processes, thereby ensuring a higher degree of legal certainty and improving the accessibility of legal texts, the results of their analysis and relevant legal information for both professionals and the general public.

As part of work package KPB4, the deliverable 'Algorithm for legal text linking' was produced. The legal document linking system is a software module implemented in Python, designed to – when provided with a source document – retrieve the most similar Slovak court decisions. The method combines statistical metrics, semantic embeddings and the analysis of structured legal data to identify documents with similar legal content, reasoning or factual basis.

The system employs a hybrid approach based on a combination of:

- semantic text similarity,
- similarity of individual content sections of the document,
- matches in legal references,
- matches in cited legislation,
- similarity of extracted key terms.

The result is a list of documents ordered by a final similarity score to the source document.

System functionality

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

1. Preparation of input data

The system works with a structured corpus of court decisions, whereby the following may be assigned to each decision:

- the text of the decision itself,
- links to related court decisions (dockets),
- cited legislation,
- extracted key terms.

The text of the document is normalised through simple cleaning, which removes redundant spaces and standardises the text formatting.

1.1. Document segmentation

Each decision is divided into smaller text segments ('chunks'). Segmentation is carried out at paragraph level, with the system preserving the natural structure of the legal text. Individual paragraphs are aggregated into chunks of approximately the same size. This approach allows similarity to be analysed not only at the level of the entire document, but also between specific parts of the legal argument.

1.2. Filtering procedural and form-based text

Before calculating similarity, the system identifies and removes procedural or form-based sections of text that recur regularly in court decisions.

In particular, texts containing the following are filtered:

- court designations,
- procedural instructions,
- standardised wording of decisions,
- recurring legal phrases.

Very short text segments that do not carry sufficient semantic content are also removed.

This step reduces the influence of general procedural language on the resulting similarity score and places greater emphasis on the substantive content of the decisions.

2. Semantic representation of documents

The SlovakBERT language model, fine-tuned for the task of semantic similarity of sentences, is used to compute vector representations of documents and text segments.

For each document, the following are created:

- a global embedding representing the entire document,
- embeddings of individual chunks.

The embeddings are normalised and then compared using cosine similarity.

3. Calculation of document similarity

3.1 Global semantic similarity

The system calculates the cosine similarity between the embeddings of entire documents. This metric represents the general thematic and semantic similarity of decisions.

3.2 Similarity of text segments

A cosine similarity matrix is calculated for all chunks in both documents. The system then identifies the most similar pairs of chunks between the documents.
Similarity is evaluated symmetrically in both directions:

- for each chunk of the first document, the best matching chunk in the second document is sought,
- for each chunk in the second document, the best matching chunk in the first document is sought.

Only the most relevant similarities above a defined threshold are included in the final score. The final chunk score is calculated as the average of the best matches.

This approach makes it possible to capture locally similar legal arguments or factual circumstances even where the documents as a whole differ significantly in scope or structure.

4. Comparison of legal references

The system compares references to related court decisions using Jaccard similarity of sets of identifiers.

Documents that refer to similar case law or identical legal cases receive a higher score.

5. Comparison of cited legislation

Cited legal provisions are represented as sets of identifiers or URLs of legal provisions. Their similarity is also calculated using Jaccard similarity, whereby the system takes into account the consistency of the legal basis for decisions.

6. Comparison of key terms

Extracted key terms are compared using weighted overlap, which takes into account:

- the presence of the same terms,
- their relative importance in the documents.

This approach makes it possible to better capture the thematic similarity of documents, even when the wording of the text differs.

7. Combined similarity score

The final document similarity score is calculated as a linear combination of several components:

- similarity of legal references,
- similarity of cited legislation,
- similarity of key terms,
- global semantic similarity of documents,
- similarity of text segments.

The similarity of chunks carries the greatest weight, as it best represents the factual and argumentative similarity of decisions.

8. Sorted list of relevant documents

Using the described similarity scoring metric, we compare the reference document to all other document in our dataset, calculating the similarity between each of them and return an ordered list of ids based on similarity score.

Technologies used

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

- Sentence Transformers for semantic embeddings,
- SlovakBERT for representing Slovak legal text,

- cosine similarity for comparing embeddings,
- scikit-learn for similarity calculations and vector operations,
- NumPy for numerical operations.

GPU support (CUDA) enables the efficient processing of large corpora of court decisions.

Source files

The project's source code and data set are maintained in a repository and are available at: https://gitlab.science.upjs.sk/altaml/document-relevance-extractor