Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220483

Scribe: A Specialized Collaborative Tool for Legal Judgment Annotation

Sid Ali MAHMOUDI [a,1], Guillaume ZAMBRANO [a], Charles CONDEVAUX [a] and Stéphane MUSSARD [a]
[a] CHROME, *University of Nîmes, France*

Abstract. *Scribe* is a legal judgment annotation platform. Its objective is to improve dataset quality for machine learning models, to make annotation task faster and easier, and to boost interactions between *annotators* and *developers*. The platform manages 3 different classes of annotation: claims, named entities and sections. The platform facilitates the expression of annotation needs by developers. Multiple annotators can quickly respond to these needs by working in parallel. The collaborative process ends when the expected model performance is reached. The platform is organized by modules, maintainable and extensible in addition to its flexibility and unified output result in JSON format. See our demo https://lawbot.unimes.fr/annotateur

Keywords. Case law, Judiciary, Dataset annotation, Legal NLP, Legal Judgment Prediction

1. Introduction

These last years, numerous Legaltech projects arose, for instance, to cite a few of them, Lexmachina, Doctrine and Caselaw analytics. The strategy of Legaltechs is clear, based on legal documents and court decisions, they employ machine learning and natural language processing (NLP) techniques to predict the outcome of a dispute: the acceptance or rejection of a claim, the quantum, the legal fees, etc.

Accordingly, organizing a massive quantity of unstructured data related to court decisions with the aid of legal experts is crucial. Based on their annotations, aiming at bringing out keywords and sentences (legal norms, names of the parties, jurisdiction, type of claim, facts, outcome, etc.), the structured data may be stored, before employing NLP models such as models for named entity recognition (NER) in order to automatically annotate all documents.

To our knowledge, there are a few researches about the design of collaborative platforms for annotating court decisions. Recently, [3] use a machine learning model to classify documents in different categories of claims and present a pseudo-platform allowing to automatically classify court decisions in those categories. However, this platform can-

[1] Corresponding Author: Sid Ali Mahmoudi, email: sid.mahmoudi@unimes.fr
This research was supported by the Occitanie region for Sid Ali Mahmoudi's PhD scholarship. The French National Research Agency is acknowledged for funding the LAWBOT project ANR-20-CE38-0013: Deep Learning for Judicial Outcome Prediction.

not be used to manage court decisions from their storage to the production of categories based on NER. [6] proposes a tool to annotate named entities with the possibility to automatically annotate court decisions. Other general-purpose annotation tools exist like Prodigy (https://prodi.gy/), but they are not adapted to the legal domain. Of course, the development of collaborative platforms to perform annotations is possible because of the emergence of new annotations techniques, see e.g. [4] and [7]. The aim of this paper is to present *Scribe*, a platform for court decisions management.

2. Overview of the platform

Scribe consists of a web application, secured by a user authentication system [1]. The main purpose of the platform is to facilitate French court decisions datasets annotation process, and communicate rapidly between legal experts and NLP developers.

System specifications. The *Scribe* prototype has been designed to get simple interfaces adapted to French legal judgments annotation tasks. It enables links creation between datasets and annotations in such way that developers can check annotation states and query the database easily. On the other hand, annotators can search for particular court decisions, annotate decisions (alone or with other experts independently on the same set of decisions), save and edit annotations.

The first step of the Scribe pipeline is composed of three tasks: claim categories, named entity recognition (NER), and sectioning the decision (header, facts, etc.). Figure 1 highlights *Scribe* use cases. Two actors can interact with *Scribe*: *Annotators* (legal experts) and *Developers*. Both of them need *admin* authorization to access the platform. For each task (claims categories, NER, and sectioning), *annotators* should be able to create, delete or update items (rows) such as type of jurisdiction and chamber for entities tables (see Figure 1); and attributes (columns) like category name or description. *Annotators* can associate a resizable dataset (can add and remove decisions as they want) to a specific task row (such as type of jurisdiction in NER task) by searching specific court decisions from the database with a dynamic number of keywords either required or excluded, wanted dataset size, and the legal judgment deliver (either Cassation, Appeal and/or First instance court). They can display the content of the decisions resulting from the search, one by one, and then select (deselect) the decisions to include (exclude). Once the decisions choice is complete, they can append them to the selected task row as the associated dataset. If the task row is already associated with a dataset, then the selected decisions are added to the existing one. *Annotators* can autonomously annotate created datasets and multiple annotators may annotate the same dataset and update annotations. They can display dataset decisions one by one in the interface annotation, delete unwanted ones, and select text spans corresponding to the correct annotation from the decision text. Once finished, they can save it, and move to another one. *Developers* may check the progress state of the annotation process continuously and can download datasets at any moment in standardized format on the homepage for easy access). The creation of a new row in a given task is a dataset query, for instance creating the row ”claims of non-recoverable charges, the article 700” in the table claims categories. After that, either *developers* or *annotators* can associate a dataset (decisions list) to this row (search by keywords and by standard jurisdiction). Once done, *annotators* can save annotations of this dataset, and the *developer* can simultaneously follow the annotation progress and download the dataset.

Modules. *Scribe* is composed of four main components, as shown in Figure 2: database component, annotation component, download component, and the user authentication component.

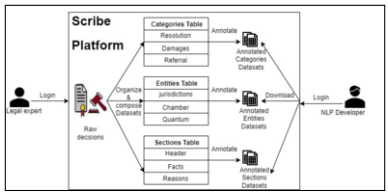

Figure 1. General overview of *Scribe*

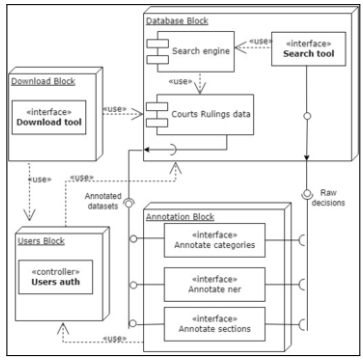

Figure 2. *Scribe* components diagram.

The database component is responsible for saving data (raw decisions, annotations, tasks classes and users) and for the database exploration with a search engine used when composing datasets. *MongoDB* is a Data Base Management System *DBMS* which is a *NoSQL* database that allows the structure of the document to be modified [2], and to handle big data.

The annotation component contains three sub-components, each one dedicated to its specific task (discussed in Section 2) and contains its suitable form. For example, in the NER form, the user selects an entity from the decision text body then all occurrences are automatically selected. In the sectioning form, the user has just to select a decision part that represents the section. As a decision can include multiple claims of the same category, the form of this task allows the *annotator* to create dynamic similar forms. The three sub-components consume dataset raw decisions (obtained from the search step in the database component) and provide their annotations (also stored in the database). *Django* is used to implement the backend logic and *Reactjs* for the frontend. *The download component* enables developers to check the available datasets and the annotation progress state (both raw and annotated datasets may be downloaded).

The authentication component manages user accounts and login, only authorized people can enter the system and read data. For more explanations, see the demo `https://lawbot.unimes.fr/annotateur`.

Data. A record in *MongoDB* is a document, which is a data structure composed of field and value pairs, documents are similar to JSON objects [2]. *MongoDB* simplifies annotation saving process and getting datasets in a standard format. It is tolerant about document keys, we can omit some in the documents and place them in others. Figure 3 represents the diagram of the database. There is a collection of system tasks, each task can have multiple sub-tasks (a dataset may be associated with each sub-task). Finally, from raw decisions, *annotators* save their annotations (dataset creation), accordingly each decision of an annotated dataset is associated with an annotation object that contains the annotations array, all elements of this array follow this format: text (annotated span of text), label (class of the text), start position, and end position.

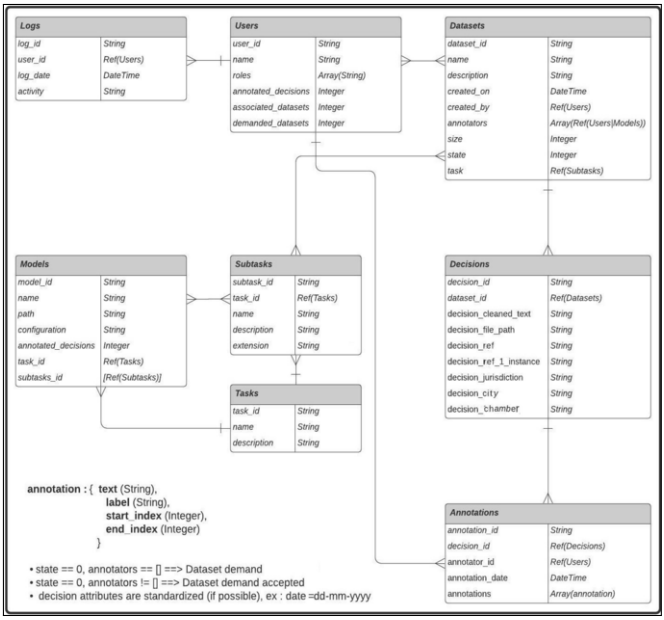

Figure 3. *Scribe* data architecture.

3. Conclusion

In this paper, a collaborative annotation web application has been proposed. This tool is dedicated to the management of court decisions for teams of legal NLP annotators and developers. As a future improvement, it should allow deploying trained models [5,8] and automating annotations.

References

[1] Barkadehi M H, Nilashi M, Ibrahim O, Fardi A Z, Samad S. Authentication systems: A literature review and classification. Telematics and Informatics. 2018;35(5), 1491-1511.

[2] Chauhan A. A Review on Various Aspects of MongoDb Databases. International Journal of Engineering Research & Technology (IJERT). 2019;8(5).

[3] Hammami E, Faiz R, Akermi I, Rosenthal-Sabroux C, Gargouri F, Arduin PE. A Dynamic Convolutional Neural Network Approach for Legal Text Classification, *Information and Knowledge Systems. Digital Technologies, Artificial Intelligence and Decision Making*. 2021, Springer, p. 71-84.

[4] Libal T. A Meta-level Annotation Language for Legal Texts. In Dastani, M., Dong, H. and van der Torre, L., *Logic and Argumentation*, Springer International Publishing. 2020;p. 131-150.

[5] Mahmoudi S, Condevaux C, Mathis B, Zambrano G, Mussard S. NER sur décisions judiciaires françaises: CamemBERT Judiciaire ou méthode ensembliste?. 2022 Jan. In Extraction et Gestion des connaissances EGC'2022.

[6] Tamper M, Oksanen A, Tuominen J, Hietanen A, Hyvönen E, *Automatic Annotation Service APPI : Named Entity Linking in Legal Domain*. 2020; Extended Semantic Web Conference.

[7] Wyner AZ, Katz WP. A case study on legal case annotation, *Frontiers in Artificial Intelligence and Applications*.2013 Jan, 259:165-174.

[8] Zhong H, Xiao C, Tu C, Zhang T, Liu Z, Sun M. How does NLP benefit legal system: A summary of legal artificial intelligence, *arXiv preprint*, arXiv:2004;12158, 2020.