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

doi:10.3233/FAIA220481

Toward an Integrated Annotation and Inference Platform for Enhancing Justifications for Algorithmically Generated Legal Recommendations and Decisions

Yi-Tang Huang⁋ Hong-Ren Lin‡ Chao-Lin Liu†
National Chengchi University, Taiwan
{108753132⁋,109753156‡, chaolin†}@nccu.edu.tw

Abstract. We introduce our workflow that integrates the steps of annotation and classification, and hope that the end products are helpful for improving the justifications for legal reasoning and for recommending similar civil cases.

Keywords. technology-assisted annotation, annotation and prediction and inference, system integration, machine learning, natural language processing

1. Overview

Convincing justifications strengthen the usability of the legal recommendations and decisions that are produced by algorithmic computations [1][3]. A legal informatics system may offer similar cases for preparing cross-examinations in courts, and may even attempt to predict the sentences against the defendants. Such assistive systems, which are constructed by the machine learning (ML) approaches, typically rely on training data to learn to select the recommendations and decisions. An ML-based predictive procedure that aims to offer satisfactory recommendations and decisions would be more useful if we can associate their outputs with appropriate supporting evidences.

We believe that such supporting evidences require at least a few high-quality annotated data for training the predictive procedure. Given a collection of original judgment documents, we use existing tools for lexical, syntactical, semantic, and even pragmatic analysis to mark the texts. Human experts can verify and correct the raw annotations. Our system also allows the annotators to read, find and mark the statements for high-level legal factors for specific categories of lawsuits. The annotated data will be used to train a new generation of tools, hopefully improving the quality of future annotations.

Currently, we use the open judgment documents of the courts in Taiwan in our system. We believe that the system architecture can be adopted by systems of legal informatics in any other languages. If the proposal is accepted, we will show the current operations of the annotation system both on site and online during the Conference.

2. Architecture and Raw Data Selection

Figure 1 shows the system architecture for the integrated system. To modularize our presentation, we show the architecture in four blocks, which are indicated on the left margin in Figure 1.

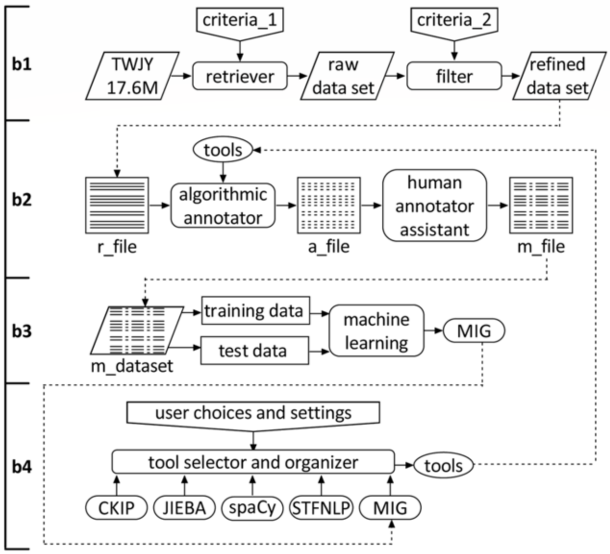

Figure 1. System architecture.

The block **b3** shows that we need an annotated dataset, **m_dataset**, to support the machine-learning approaches to train a classifier, **MIG**. We split the annotated dataset into two parts for training and test to produce the classifier.

In order to obtain the annotated dataset, we first need to extract appropriate files from a large collection of judgment documents. In our current work, we use the open data repository of the Judicial Yuan of Taiwan as the main source. We show this repository as **TWJY** in block **b1**. There are about 17.6 million documents in TWJY as of April 2022. Assuming that our research focuses on a specific category of cases, say burglary. We can use "burglary" as a keyword in **criteria_1**, and let the **retriever** extract documents that are related to burglary from TWJY. The resulting **raw data set** usually includes some documents that do not really fit the research requirements from the legal informatics perspective. Hence, we would consult the expertise of legal experts, and use the **filter** to remove some files from the raw data set based on **criteria_2** to produce the **refined data set**. After this step, our system offers an interface via which a human expert can select to read a specific document to determine whether to annotate a file in the set.

Depending on the research issues of the researchers, the size of refined data set may not be large, even though we have millions of documents in TWJY. TWJY includes documents for lawsuits of all possible criminal, civil, and other special cases of between 1996 and April 2022. Some types, e.g., burglary and gambling, can be relatively common, but some are relatively infrequent, e.g., verification of presumption of paternity.

3. The Annotation Procedure

The block **b2** shows the main steps for the annotation task. Usually, the human annotator chooses to annotate one file, **r_file** (“r” for raw and refined), from the refined data set at a time. In fact, depending on the needs of the research, an annotator may just need to examine some specific parts (sections) of the documents, e.g., the *facts and reasons*, and do not need to inspect the whole judgment document. If you may find the “理由” in eighth line of the center window of Figure 2, that is the Chinese word for reasons.

Figure 2. A sample of a_file and the current human annotator assistant.

Figure 2 shows a sample of **a_file** (“a” for annotated) and the current implementation of the **human annotator assistant**. In the center of Figure 2 is a sample of a_file. The selected r_file was annotated by the **algorithmic annotator**. The algorithmic annotator uses the tools that are set up in the **tools** component (to be explained shortly). In this example, the tools component is just the *named entity recognition* (NER) component of the CKIP toolset [2]. It is our software that colors the outputs of the CKIP-NER by the types of entities, and coverts the file format for the reading and annotation interface. The current CKIP-NER identifies seven categories, as is shown on the upper right corner of Figure 2, where “ORG” and “GPE” refer to organizational and geographic entities, respectively. The categories of the words in the text are indicated by their colors.

We can equip the human annotator assistant with multiple functions. The most basic ones include allowing the annotators to *create*, *update*, and *delete* the annotations that are suggested by the algorithmic annotator, whose suggestions may not be perfect.

It is evident that the seven categories of entities that the CKIP-NER attempts to identify is insufficient for the needs of legal reasoning. For the cases of burglary, it might be preferable to annotate the places of the events, the thief, the stolen objects and their values, and perhaps whether the thief is armed. An annotator can add such additional categories in the box on the upper right corner of Figure 2, where we put a “TEST” box as an example. An annotator can add a new tag, then chooses texts in the center window, and labels the chosen texts with the new tag. More than one tag can be added in a session.

The RELabel button leads the annotators to enter an interface for editing regular expressions to define patterns. A pattern can represent an established way to express a legal notion in the text, and will be assigned a tag. Defining patterns for legal notions in regular expressions allows the software to identify the patterns automatically for us. A more complex system for annotation could offer a mechanism for the annotators or the research team to maintain and share useful domain-dependent regular expressions. The RELabel button in Figure 2 offer the annotators to try their instincts for automatic annotations on the fly.

The Download button allows the annotators to save the results of the current annotation, i.e., **m_file** in Figure 2. A previously annotated file can also be uploaded to the annotation system again so that the annotators can revise the annotations. To protect the previously annotated results from being overwritten, the algorithmic annotator will be suppressed for reloaded files.

4. Progressively Powerful Models

We gather a sufficient number of m_files to obtain the m_dataset in b3, and run whatever types of machine learning procedures we plan to. In **b4**, we show that we may use whatever software tools as the algorithm annotator, including JIEBA, spaCy, Stanford NLP tools (STFNLP in Figure 1).

Whenever possible and desirable, we may use our own model, **MIG**, as one of the possible tools to annotate the documents. This is not only possible but should be more reasonable when the size of the refined data set is large such that it is unreasonable and inefficient to expect human experts to inspect and annotate all of the data files all at once.

Assume that we have N files in the refined data set and that N is large. We can split the refined data set into p parts, each having $\frac{N}{p}$ files. Assume further that, we choose one of the tools in b4, but not MIG, as the algorithmic annotator to annotate an installment of $\frac{N}{p}$ files in the first iteration. The annotator may examine, check, and revise annotations of these $\frac{N}{p}$ files. We can use these files to train an MIG, and use this MIG in the next run.

Assume that the MIG can perform better than the other choices, because it is directly trained with our annotated data, but the other tools do not have this privilege. Hence, starting from the second iteration, we choose MIG as the algorithmic annotator. In the second run, we use the second installment of $\frac{N}{p}$ files. Since MIG is a better model, we expect the quality of the annotations in the a_file is better than before. As a result, it is easier for the human annotator to create, update, and delete labels this time. This type of improvement can continue in the following runs at least on average. Hence, splitting N files in the refined data set into parts can make our annotation process more efficient.

We have assumed that the MIG we obtained in the first iteration will outperform the other tools. If this is not true, we may wait until MIG can outperform others before we switch to our own MIG in b4. If MIG continues to underperform, we probably should improve our machine learning procedure in block b3 first.

5. Concluding Remarks

We have used our own annotation tools for our work on classifying statements by their functions in judgment documents for civil cases. Statements for civil cases typically are more indecisive than those for criminal cases, so the availability of human annotations is more helpful and necessary. We attempt to identify the statements of the requests of the plaintiffs, of the responses of the defendants, and of the viewpoints and decisions of the judges. More importantly, we attempt to automatically find the conflicting issues between the plaintiffs and the defendants. Based on these results, we hope to find “similar” civil cases. The results of this application-oriented research remain promising.

Acknowledgements

This research was supported in part by the project 110-2221-E-004-008-MY3 of the National Science and Technology Council of Taiwan.

References

[1] Atkinson K, Bench-Capon T, and Bollegala D. Explanation in AI and law: Past, present and future. *Artificial Intelligence*. 2020; 289:103387.
[2] Li P-H, Fu T-J, and Ma W-Y. Why Attention? Analyze BiLSTM Deficiency and Its Remedies in the Case of NER. *Proc. of the 33rd AAAI Conference on Artificial Intelligence*. 2020; p. 8236–8244.
[3] Mumford J, Atkinson K, and Bench-Capon T. Machine learning and legal argument. *Proc. of the 21st Workshop on Computational Models of Natural Argument*. 2021; p. 47–56.