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

doi:10.3233/FAIA220484

An Interactive Natural Language Interface for PROLEG

Ha-Thanh Nguyen [a] Fumihito Nishino [a] Megumi Fujita [a] Ken Satoh [a]
[a] *National Institute of Informatics, Japan*
nguyenhathanh@nii.ac.jp

Abstract. PROLEG is a famous computer program supporting attorneys in the legal inference process. However, the input of this system is expressed in Prolog, which most lawyers are not familiar with. This technical barrier is a serious problem for using PROLEG in the real legal context. A natural language interface is one of the solutions to this problem. We have developed a prototype of such an interface. This paper describes the prototype and its current performance. The prototype translates input facts into Prolog expressions following PROLEG syntax. The system consists of three main modules, (1) natural language perceiver, (2) PROLEG reasoner, and (3) inference explainer. In addition, we analyze the performance of the prototype and identify existing issues and discuss possible solutions.

Keywords. interactive interface, natural language, PROLEG

1. Introduction

Legal inference is an important task for lawyers. The legal inference process consists of two main phases: collection of the facts and inference from the facts. In order to reach a legal goal, for example exculpation, liability proving, or sentencing, the attorney collects facts and infers the legal conclusions from the facts. However, legal inference is a complex process and attorneys can make errors in the legal inference process. There are several reasons for this. First, the legal inference process has many steps, and it is difficult to check all steps. Second, there are many rules of law, and it is difficult to remember all rules of law. Third, the legal inference process is time-consuming and attorneys often have to make decisions under time pressure.

PROLEG [1,2] is a legal reasoning system based on the Japanese "theory of presupposed ultimate facts" (called "Yoken-jijitsu-ron" in Japanese, the JUF theory [3], in short). Incomplete information is a common issue of many legal cases. In practice, judges usually have to make a decision even if they do not have all information they need. In such a situation, the JUF theory is used for decision-making. The presupposition of ultimate facts is the key idea of the theory. An ultimate fact is a fact that cannot be further explained or justified. These facts are not necessarily true, but they are assumed to be true for the sake of argument. For example, in a criminal case, the defendant is assumed to be innocent until proven guilty.

PROLEG was proven to be useful in actual legal cases. However, the input of this system written in logic programming can be only in the form of logical formulas. This

Figure 1. General architecture of the system.

creates a technical barrier for attorneys and judges who are not familiar with logic. In order to make PROLEG more user-friendly, we propose an interactive inference system allowing users to input natural language. The system contains three modules that receive fact descriptions in natural language, convert them into PROLEG formula, operate the logical reasoning process and return the output along with its visual explanation.

2. System Description

2.1. General Architecture

The general architecture of the system is described in Figure 1. There are three modules which are (1) Natural language perceiver, (2) PROLEG Reasoner, and (3) Inference Explainer.

The natural language perceiver is used for processing the fact description in natural language, extracting the important information, and constructing the corresponding PROLEG formulas. This module is implemented using a combination of a translation deep learning model and supportive heuristic rules. We use BART [4] for the translation model with different strategies:

- **naive end-2-end translation:** we simply train a BART model with our PROLEG pair samples without any further tweaking.
- **translation and correction:** embedded behind the translation model in the framework, the correction model is trained to map the errors back to the originals. These errors are generated by random heuristic rules (such as capitalization, splitting, removing, adding, replacing).
- **argument recognition**: we train a BART model to recognize the arguments for PROLEG formulas instead of directly translating the natural language into PROLEG formula.

Evaluating the outcomes of the three approaches, it was found that the argument recognition approach had the best performance. The reason is that the model does not need to remember the whole syntax of the logical formula but only to detect the correct arguments. With the current limitation in the number of training samples, the translation model sometimes wrongly detects the date-related arguments, we come up with a workaround to prevent this by regular expressions.

The extracted PROLEG formulas are then used as input to the PROLEG reasoner, which is implemented in Prolog. The PROLEG reasoner contains a manually pre-defined set of PROLEG rules and performs the inference by using these rules. The output of PROLEG reasoner is the result of the inference as well as the intermediate reasoning steps.

The final module is the inference explainer which is used for the explanation of the inference result. The explanation is presented in the form of an interactive graph, which

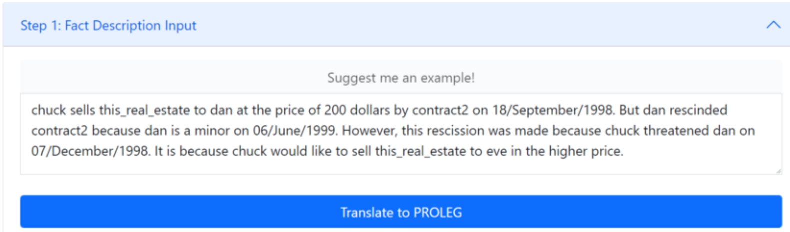

Figure 2. Fact Description Input.

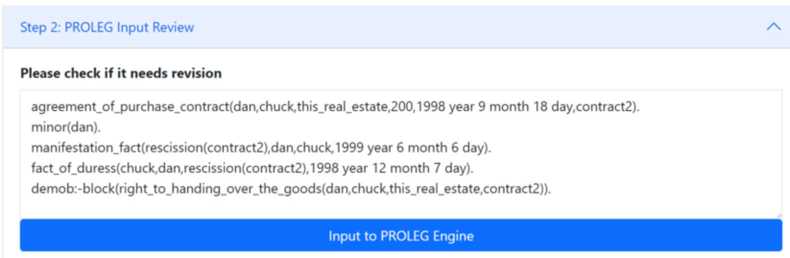

Figure 3. PROLEG Input Review.

the user can use to trace the inference step-by-step and learn the importance of PROLEG rules in each step. This explainer module reduces the complexity of understanding the PROLEG reasoner results and makes it understandable for lay users.

2.2. *User Interface*

Fact Description Input. The user interface of this step is presented in Figure 2. The user can input a natural language fact in the text area and click on the submit button. The output of the natural language perceiver is PROLEG formulas for the input sentences. For the demonstration purpose, the user can choose to use sentences suggested by the system and modify them if necessary.

PROLEG Input Reivew. The user interface of this step is presented in Figure 3. The translated PROLEG formulas are in the text area, the users can validate the formulas and make modifications if necessary. After the formulas are verified, users can input them into the PROLEG Engine.

PROLEG Inference Graph. The user interface of this step is presented in Figure 4. The PROLEG inference explanation is presented in the form of an interactive graph. Each node in the graph represents a PROLEG rule. If the node condition is satisfied, the node is presented in green, otherwise orange. By clicking on the node, the user can investigate the satisfiability of the rule condition and the reason for such results. This will help users to understand the PROLEG rules and improve their intuition about the underlying reasoning process. In the in-development version, the user can also click on the "step-by-step" button for a detailed explanation of the inference path.

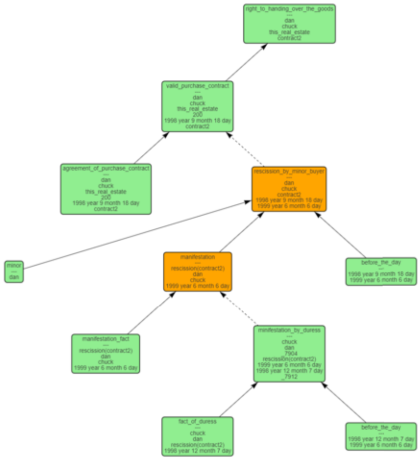

Figure 4. PROLEG Inference Graph.

3. Conclusions

This paper describes the development of an interactive inference system for the PROLEG core language. The system is designed to reduce the burden of writing logical formulas for lay users and provide an interactive explanation of the inference result to help users understand the underlying reasoning process. This system is an important link between the knowledge representation research and the real-life application for lay users, which will help to expand the application of PROLEG in the field of intelligent reasoning. Since the feasibility of the system has been confirmed by design, the next step of this work is to improve the system through data preparation and method enhancement for each module.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers, JP17H06103, JP19H05470 and JP22H00543 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4, Japan.

References

[1] Satoh K, Kubota M, Nishigai Y, Takano C. Translating the Japanese Presupposed Ultimate Fact Theory into Logic Programming. In: Proceedings of the 2009 conference on Legal Knowledge and Information Systems: JURIX 2009: The Twenty-Second Annual Conference; 2009. p. 162-71.

[2] Satoh K, Asai K, Kogawa T, Kubota M, Nakamura M, Nishigai Y, et al. PROLEG: an implementation of the presupposed ultimate fact theory of Japanese civil code by PROLOG technology. In: JSAI international symposium on artificial intelligence. Springer; 2010. p. 153-64.

[3] Ito S. Lecture series on ultimate facts. Shojihomu (in Japanese). 2008.

[4] Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020. p. 7871-80.