Rule Learning from Text for Legal Question Answering

ABSTRACT

Law is a system containing rules and regulations that binds a people. Legal rules and regulations are usually expressed in domain specific terminologies which are presented in textual form. Its expression is not in machine understandable format for legal reasoning to infer new knowledge or determines if a course of action aligns with the law. Similarly, in order to conceive the rule layer of the semantic web vision in line with the W3C recommendation, in this paper, we present a rule learning technique for learning legal rules for legal question answering, where we learn rules from a collection of instance level triples to infer new rules which can be applied to facts to reason with to arrive at an answer. We explore the natural language processing tool to extract instance level triples from legal textual data and applied RUMIS tool on the extracted triples to produce nonmonotonic rules which are then translated and expressed in Semantic Web Rule Language for legal reasoning in answering legal questions. The research output shows promising results with respect to rule learning for legal reasoning for question answering.

Keywords: Law, Semantic web, Rule learning, nonmonotonic rules.

1. INTRODUCTION

With the rising interest in conceiving the different layers of the semantic web vision, in which the ontology layer has grown to some degree of maturity based on the W3C recommendations like the Web Ontology Language OWL. Recent research focus on the rule layer of the architecture. Though, different solutions have been proposed to solve this problem but lack the straightforward answers due to various obstacles. One of the challenges is that of the evaluation principles such as the closed-world assumption which is often not adopted in ontologies but are found in rule languages. Rules are built for some specific tasks which can be applied independently of the ontology whereas in other cases application need rules to extend the expressivity of web ontology language which require a combination of ontology and rules for reasoning in solving problem. In this paper, we present a rule learning technique for learning legal rules for legal question answering, where we learn rule from a collection of instance level triples to infer new rules which can be applied to facts to reason with to arrive at an answer.

Law is a system containing rules of conduct created and implemented by social or governmental organizations to govern the behaviour of people. Laws are intended to protect the individuals and their properties from undesirable conflict from others. They are often expressed with domain-specific terminology and are conveyed in textual form. Its expression and presentation do not provide a standard structure for a machine to use and reason with. Moreover, capturing legal rules and expressing them in machine understandable format for legal reasoning is not a linear task; whereas applying formal reasoning techniques require the processing of formal conditions to infer newer knowledge, or determine whether the observed action aligns with the condition or not.To apply the law to a given legal case, human expert needs to have the right knowledge of the law and its application in taking judicial decision. In the same way for an information system to be able to perform legal reasoning, such tools must have the required legal knowledge and its application to satisfy the same property. Like the human inductive reasoning process, computing machines need to learn and apply knowledge learned in a typical scenario to others.

In recent times Machine learning (ML) approach have been found useful in predicting solutions without being explicitly programmed. ML can identify patterns of data points as a way of learning new knowledge, but its learned knowledge cannot be expressed in a form that could be understood by human. The difference between such systems with human is the lack of transferability and interpretability. However, the learned knowledge can only be applied in a specific scenario in which it was trained.

2. PROBLEM STATEMENT

The legal environment has experienced a great intensity and influx of information which is due to the rising number and variety of information sources. Inspired by the huge volume and type of data, the advances in question answering, textual entailment and information extraction have led to the adoption of the so-called ontology technology [18, 19]. Ontologies contains a collection of semantic triples in the form of (subject, predicate, object) based on the RDF data model [16, 17]; where the ‘subject’ and ‘object’ define entities and the predicate a relation connecting the entities. The triples hold facts about the world which can also be expressed by means of unary and binary predicate.

Moreover, since ontologies are automatically constructed or instantiated, they may contain incomplete information due to the incomplete and bias nature of textual sources in whichsome things are not stated explicitly. Hence, ontologies work based on Open World Assumption (OWA) which means treating what is not known to be true or false to be unknown information. Completing such information is of crucial importance for the application of rules for legal reasoning.

3. LITERATURE REVIEW

Recent studies have established different techniques for completing missing links in knowledge graphs. For missing links predication two major methods (statistics based and logic-based approached) have been often employed by researchers [15].

On the one hand, the motivation of the statistical based approach is to build a model with latent features that may not explicitly observable from the original data [25]. The general idea is to determine a statistical correlation between the objects with respect to hidden features. These approaches employ the inductive logic programming (ILP) on relational association rule mining. On the other hand, the logic base approach is more interpretable compared to the statistical [20]. They identify observable patterns to determine new edges in a knowledge graph. Some of the logic-based learning tool adopt inductive logic programming, relation learning and nonmonotonic rule mining systems. While both statistic and logic-based approaches work with data inside the knowledge graphs, text-based approach work with data outside of a knowledge graph such as Wikipedia to identify relational objects. Others learn lexical relations between entities [24].

Rule learning has been studied in [12, 13] to support the challenge of knowledge graph completion. [12] present an abductive reasoning task that depends on the transformation from an inductive logic programming task. The approach learns nonmonotonic rules from a complete dataset based on Close World Assumption (CWA) that takes in what is not known to be true to be false. [13] proposes a nonmonotonic ILP system known as XHAIL which integrates an abductive logic programming technique that has a direct route to machine learning. It works with incomplete theories thereby delivering semantics for negation as failure. [14] describes a machine learning approach for Horn rule learning in the presence of incompleteness in a hybrid environment. The technique combines description logic and disjunctive Data-log for inductive view definition as well as integrity theories in a relational database with ILP. [20] present an all-encompassing overview of completing the missing link representation learning with knowledge graphs. The approach applied statistical relational learning technique for learning a large knowledge graph to predict new facts about the world. This involve predicting new edges in a graph. The study also discussed different related techniques for encoding huge number of edge types. A rule mining technique have been developed and applied on a large knowledge base [21]. The approach applied ILP technique to mine logical rules from knowledge bases. It learns associative rules from RDF data. The primary drawback of the approach is that rules are limited to Horn clauses in their logical expression. That means that existential quantifiers or disjunctions are not allowed. Exception rule evaluation based on exceptionality measure is described in [22]. For decision making, rules are presented in the form of negation and positive association. The approach applied a Fussy based method which speed up the mining process. In exception rule mining, the state-of-the-arts techniques are direct and indirect. Direct cases are highly subjective whereas indirect applies knowledge derived from a set of rules which are usually strong rules. Exception rules contradict or change the knowledge. [23] propose a rule mining technique for mining rules and applies rule refiner to refine mined rules by adding negated atoms. The approach works by first learning a set of Horn like rules which are then revised by adding negated atoms to the body of the rule in order to account for exceptions. It extracts exception enriched nonmonotonic rules from incomplete knowledge graph. Here, all binary facts are projected as unary thereby applying propositionalization technique to transform them into Horn rules. The rules are then augmented with negated atoms.

With the rising interest in conceiving the different layers of the semantic web vision, in which the ontology layer has grown to some degree of maturity based on the W3C recommendations like the Web Ontology Language OWL. Recent research focuses on the rule layer of the architecture. Though, different solutions have been proposed to solve this problem but lack the straightforward answers due to various obstacles. One of the challenges is that of the evaluation principles such as the closed-world assumption which is often not adopted in ontologies but are found in rule languages. Also, combining rules and ontologies raises the issue of undecidability. [1,4] implemented a hybrid approach in supplementing OWL with rules. [1] adopt the Semantic Web Rule Language SWRL rule Tab, a Protege plug-in for the acquisition of rules. SWRL rules Tab help to translate the conjunctive rules into Jess knowledge representation as facts and rules in the form of IF...THEN rules. The new facts realised from inference are then moved back to protege-OWL as OWL knowledge. [2] used Protege OWL, Racer and Jess with an intent to integrate OWL-DL and SWRL rules components together for interoperability and inferentially to get all the inferences. Golbreich applied a pragmatic approach for reasoning over ontologies and rules with respect to the Semantic Web standards using existing tools for implementation [3]. The application of rules to ontologies has been found useful in reasoning about facts triggered from terminological information defined in the ontology [4]. A prototype system HD-Rules were implemented in [5], which integrates description logics with logic programming. The approach worked with existing OWL ontology reasoners and an XSB prologue for ontologies and rules reasoning, respectively. Yang and Cheng presents a DLclog hybrid formalism which integrates Description Logics and Logic Programming for the Semantic Web [6]. The approach is an extension of the Rosatis DL + log that accepts the occurrence of negative dl-atoms in the antecedent part of the rule.

4. METHODOLOGY

In this section, we present our rule learning methodology in a step-by-step fashion leading to the creation of new rules for the legal question answering task (See Figure 1). First, we apply Natural Language Processing (NLP) tool [7] to extract instance level triples from our input legal text which serves as input data to the RUMIS tool [8] (see Subsect. 4.1). Then, the RUMIS tool is applied on the extracted triples to produce nonmonotonic rules under the OWA (see Subsect. 4.3). The nonmonotonic rules are then expressed in Semantic Web Rule Language (SWRL) for legal reasoning in order to answer legal questions. The step-by-step process includes the extraction of instance level triples, some manual intervention, and the application of RUMIS for learning rules for legal reasoning.

4.1 DATASET

The dataset for this research was retrieved from the United States Multi-state Bar Examination (MBE) material, which was provided by the National Conference of Bar Examiners (NCBE). The original material contains 200 bar exam questions and are organized in the form of background information and four multiple-choice answer statements. An example of the original question with answer statements is:

After being fired from his job, Mel drank almost a quart of vodka and decided to ride the bus home. While on the bus, he saw a briefcase he mistakenly thought was his own, and began struggling with the passenger carrying the briefcase. Mel knocked the passenger to the floor, took the briefcase, and fled. Mel was arrested and charged with robbery. Mel should be
a. acquitted, because he used no threats and was intoxicated.
b. acquitted, because his mistake negated the required specific intent.
c. convicted, because his intoxication was voluntary.
d. convicted, because mistake is no defense to robbery

The spectrum of the coverage of the bar examination includes Constitutional, Contract, Evidence, Real Property, Torts, Civil Procedure and Criminal law and Procedure. Hence, the bar exam is broad and vast in exploring the examinee's understanding of the law as it relates to the application of the law in the United States.

4.2 INSTANCE LEVEL TRIPLE EXTRACTION

First our approach extract instance level triples from the source text which serves as input to the RUMIS tool. Manually extracting semantic information (instance level triples) is time-consuming and error prone. Here, our interest is to extract semantic triples from the input text and use them to feed the RUMIS tool. Triples are a collection of three entities that represent a statement derived from the textual information and which are expressed as s-p-o (Subject-Predicate-Object). For the task of triple extraction from the input text, we employ a Natural Language Processing (NLP) technique to extract semantic triples [7].

With this implementation, the schematic information for these relationships will not require specification in advance. For example, Jill is charged with robbery, would create semantic triples (s=Jill; p=charge with; o=robbery). We implement the Stanford CoreNLP to perform tokenization, part-of-speech tagging, lemmatization, named entity recognition, dependency parsing, Stanford openie and natural logic. The system takes a sentence and chunks it into a set of entailed clauses, and each of these clauses is then shortened to produce a set of sentence fragments. These fragments are then segmented into the Stanford NLP triples. Here, the sentence splitter and tokenizer are used to identify sentence and word boundaries. The tokenizer chop sentences into pieces called tokens. The outcome of annotation set serves as input for the next processing step.

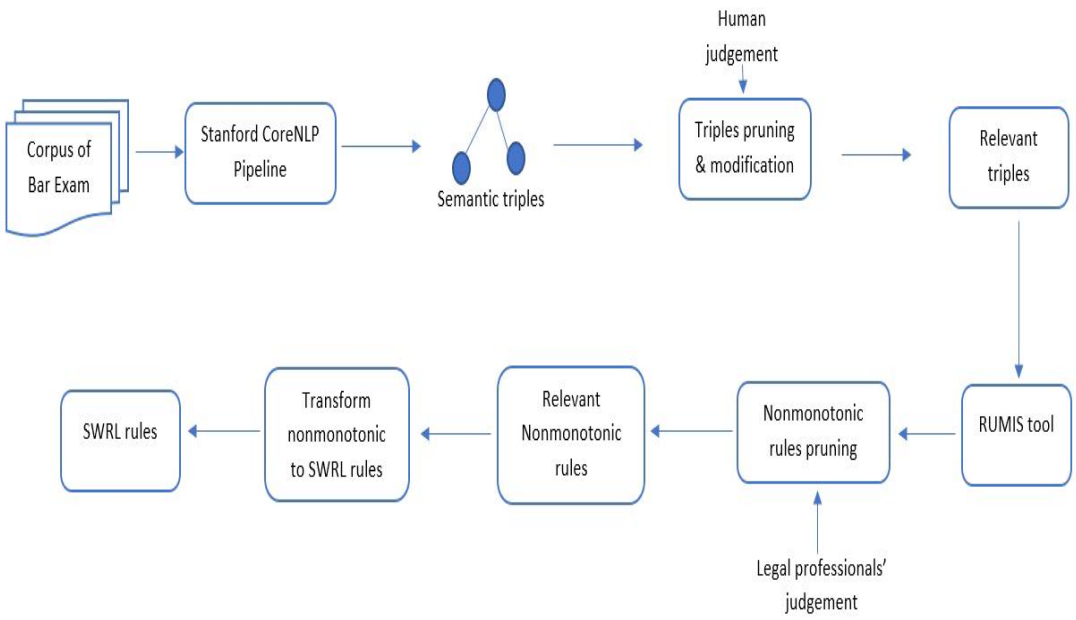

Fig. 1. Rule learning for legal question answering.

The POS tagger annotate each word with its respective syntactic tag based on context, thus describing whether a word is a noun, verb, adjective and so on. It works by taking in a string of words, process them by tagging them with the corresponding part of speech and presents a list of tagged words. For example, the sentence John forcefully collect Jane's laptop. Will be tagged as (*John/NNP forcefully/RB collect/VBP Jane/NNP's/POS laptop/NN*) Where NNP means proper noun singular, RB adverb, VBP verb non-3rd person singular present, POS possessive endings, and NN noun singular. Then the lemmatizer is applied to reduce words to their base form thereby removing all the several inflectional forms of the words to be analysed. The lemmatizer is a component for structuring words to their simplest component (lemma, prefix, affix, etc). For example, the words (learned, learn, learnt, and learning) has the base form learn known as lemma which can be looked up in a dictionary.

Dependency parsing, Natlog and Openie is then applied to produce instance level triples. Parsing defines the syntactic roles in the various input sentences which describes the grammatical relationship existing between subject and object. Applying parsing upon the input text enables the identification of the syntactic roles. Take for example, the sentence John bought a book." The syntactic role here is `bought', `John' is the subject and the `book' is the object. The Natlog component decomposes the input text into chunks of lexical entailment relations thereby connecting the premises to the hypothesis [9]. The Openie extracts relational tuples, specifically binary relations from a natural language text [10].

4.3 TRIPLE PRUNING AND MODIFICATION

Apart from the normal difficulty inherent in processing legal information such as the challenge of extracting all the possible circumstances in each of the criminal law

questions, most legal questions contain implicit information. In the triple extraction phase, there are some other problems we encountered in using the Stanford CoreNLP pipeline to extract semantic triple information from the natural language input text. For example, the extraction of complex, modifiable, relevant, and irrelevant triples. As such, some of these challenges require manual intervention in order to produce more accurate instance level triples for running the RUMIS tool to generate legal rules.

Complex subject and object contain a noun phrase with a combination of any of words, phrases, or clauses that modifies it. While compound subject and object contain two or more noun phrases connected by coordinating conjunction. For example, the triple (*s=police officer periodically p=visit o=motor vehicle junkyard in town*) contains complex compound subject and object.

Amongst the extracted triples, some are relevant, and some are irrelevant. Relevant triples contain simple subject, predicate, and object, and conveys an important information. For example, the following triple (*s=Mel, p=knock, o=passenger*) is relevant. Reoccurring or duplicate triples and triples that do not convey meaningful information are regarded as irrelevant. They are triples we cannot make sense off.

In other cases some of the complex or compound triples are modifiable as well as the cases where direct or indirect object is not captured. For example, given the sentence (*Joe admired Martys wristwatch*) the tool only captures (*s=Joe, p=admire Martys, o=wristwatch*) as the 'subject-predicate-object'. This is the case where the system is failing to capture the exact predicate admire. Hence, we manually modify the modifiable cases and discard duplicate cases while sustaining relevant ones. For example, the triple output (*s=Joe, p=admire Martys, o=wristwatch*) from the sentence (*Joe admired Martys wristwatch*) is modifiable. As such, we modify the triple by removing the portion of predicate with 'Marty' with the direct predicate '*admire*' in the sentence.

4.4 RUMIS

RUMIS is a nonmonotonic rule mining tool. It transforms Horn rules to nonmonotonic ones based on OWA. RUMIS mine relational nonmonotonic rules from a collection of instance triples based on the OWA thereby transforming the problem into a theory revision task and applies associative rule mining technique to manage large size of knowledge graph. The approach follows three processing steps geared towards producing positive nonmonotonic rules.

- The first step takes in a knowledge graph and treat every fact as a document in which the terms are subject, predicate, object, or combination of them. For example, given the triples (*<Billy is guilty of robbery>, <Anna is guilty of murder>, <Kim take briefcase>, <Kippa kill Beth>*) can be indexed as (*is guilty of <Billy, robbery>; is guilty of <Anna, murder>, take<Kim, briefcase>, kill<Kippa, Beth>*). Here, in this example the indexing is based on predicates to retrieve the respective subjects and objects.
- Positive rule mining step computes a collection of rules in the form such that the absolute support extends the given threshold which is implemented based

on the Horn rule learning algorithm. The output is then sorted in descending order of their absolute support. For example, given the triple (*x, p, y*) from a knowledge graph *K*. Adapting the index function in step 1, can a pair *z, r* be found such that (*z, r, x*) is in *K*. This is repeated for all the triples.

- This step computes the normal and abnormal instance set for each of the rules. For example, given the knowledge graph K, if a tuple is found in K, it is identified as normal, otherwise it is abnormal.

4.5 NONMONOTONIC RULE PRUNING

The output of the RUMIS tool is a set of nonmonotonic rules (legal rules). Theserules contain a head with one atom and body with two atoms in the form as shown below:

$$\text{move\_motion}(x, y) \wedge \text{suppress\_evidence}(y, z) \rightarrow \text{suppress}(x, z)$$

However, the tool over-generates nonmonotonic rules, that is, nonmonotonic rules generated which are irrelevant for the purpose of legal reasoning.

$$[perform(x, y) \wedge dangerous\_to(y, z) \rightarrow disregard(x, z)]$$

$$[put\_explosive(x, y) \wedge kill\_person(y, z) \rightarrow kill(x, z)]$$

$$[search(x, y) \wedge be\_obtain\_as(y, z) \rightarrow find\_material(x, z)]$$

$$[place(x, y) \wedge take(y, z) \rightarrow take(x, z)]$$

Here, the last rule set $[\text{place}(x, y) \wedge \text{take}(y, z) \rightarrow \text{take}(x, z)]$ which reads as "*Person place property and property take property implies person take property*" is incorrect and such irrelevant. Hence, we manually scan through all the mined rules to remove all the irrelevant rules extracted. In the same way, the rule $[\text{perform}(x, y) \wedge \text{dangerous\_to}(y, z) \rightarrow \text{disregard}(x, z)]$ is relevant and reads as "*Person x perform an action y and y is dangerous to person z implies that x disregard z"* is useful in connecting missing links with our earlier handcrafted legal rules in [11]. Here, 192 nonmonotonic rules were generated from 2352 instance level triples extracted from the triple extraction phase. However, 52 out of the 192 nonmonotonic rules were identified as relevant and 140 as irrelevant.

4.6 EXPRESSING NONMONOTONIC RULES IN SWRL

Here, we read-in each relevant nonmonotonic rule and expressed it as Semantic Web Rule Language (SWRL) rule into our knowledge base [11, 27]. SWRL was developed to be one of the standard rule languages for the semantic web. The inserted new rules and our handcrafted (manually created) legal rules are combined together for legal reasoning. Moreover, in our triple extraction phase, triples were extracted based on our ontological structure; hence, there was no disparity in the new rules and our handcrafted rules. The new rules also contain the same form as our handcrafted rules. The nonmonotonic rules derived from the RUMIS tool contain

one atom in the rule head and two atoms in the body. For SWRL variable creation, the variables in each nonmonotonic rule is translated into SWRL.The '*factory.getOWLClass*' method was used to hold the reference to the class (see nonmonotonic rule and the translated SWRL rule).

Nonmonotonic rule: [disregard(x, z): −perform(x, y), dangerous_to(y, z)]

SWRL rule: [perform(x, y) ∧ dangerous_to(y, z) → disregard(x, z)]

In translating the nonmonotonic rules into SWRL expression, the 52 relevant nonmonotonic rules derived from the rule learning phase are transformed into SWRL rules and are inserted into our knowledge base.

5. EVALUATION

Our rule learning approach is implemented with Java. The research was conducted with randomly selected 90 MBE (criminal law) questions from our corpus of US multistate bar examination questions [15]. The rule learning algorithm was run on the 90 question-answer pairs. We adopt the K-fold cross validation technique to evaluate the performance of the tool, where K is equal to 3. As such our dataset was grouped into three distinct set (A, B, and C), of which each group had 30 questions. A trial run was carried out on the three derived distributions of the questions as in (A+B:C) to train and test the RUMIS tool. Which means that in the first instance, we train the system with the first 60 out of the 90 questions and used the remaining 30 for testing. In the second phase we had (A+C:B)questions for training the system and use the other 30 for testing. Finally, we trained the system with (B+C:A) questions and test with the first 30 questions. All of these was an attempt to evaluate the influence of any data bias in the results.

We evaluated the rule learning tool in two ways: quality and correctness, and task based. For quality and correctness, we manually evaluated the learned rules with respect to our handcrafted criminal law rules and find the learned rules suitable for our purpose. Also, we evaluated the legal rules in conjunction with legal professionals. If a rule set is said to be correct by the legal professionals, then that ruleset is extracted as correct ruleset. Assessing the performance of the combination of the handcrafted rules and the learned rules with respect to our benchmark answers in our corpus of bar examination questions; our first (A) and second (B) distributions yielded poor results. However, in the last (C) distributions 6 out of 30 questions were answered correctly.

In analysing the failing cases in the other distributions, we identified that the poor performance is due to the disparity and heterogeneity in our experimental data. Also, we observed that the experimental data (90 question answer pairs) is not sufficiently large enough for the task. In addition, we found that the learned rules are better used to augment the more general criminal law rules (our handcrafted rules). Moreover, our handcrafted rules do not cover all 90 selected questions; rather they cover only 30 questions. One of the six correctly answered questions in this case is question 107 in our source material. This question is on larceny crime. We observed

that two of our handcrafted rules with one of the mined rules were applied in answering this question. The two larceny rules applied are rule 5.1 and rule 5.2.

$$bring\_property(?y,?p) \wedge keep\_property(?x,?p) \wedge larceny(?l) \rightarrow intend\_to\_commit(?x,?l) \quad (5.1)$$

Rule 5.1 reads as follows: person y bring property p and person x keep property p impliesthat person x intends to commit larceny.

$$take\_property(?x,?p) \wedge keep\_property(?x,?p) \wedge intend\_to\_commit(?x,?l) \wedge larceny(?l) \rightarrow has\_committed(?x,?l) \quad (5.2)$$

Rule 5.2 read as follows: person x take property p and keep property p and intend to commit larceny implies that person x has committed larceny.

For the mined rule is:

$$tell\_person(?x,?y) \wedge bring\_property(?y,?z) \rightarrow take\_property(?x,?z) \quad (5.3)$$

Which reads as follow: person x tell person y and y brings property z implies that personx take property z. For example, from question 107, the following ABox information in Figure 2amongst others were extracted and used to populate the ontology.

$$\{bring\_property(Roy, television-set), keep\_property(Grace, television-set), tell\_person(Grace, Roy), use\_person(Grace, Roy), carry\_propert(Roy, -set), be\_accessory\_to(Grace, Roy), larceny(larceny)\}$$

Figure 2 ABox information from question 107

Here, using the ABox information, the handcrafted rule *5.1* and the mined rule *5.3* will fire to produce the respective outputs:$\{intend\_to\_commit(Grace, larceny)\}$ and $\{take\_property(Grace, television-set)\}$. The conclusions of rules *5.1* and *5.3*, along with the ABox assertion$\{keep_property(Grace, television-set)\}$ are then used as input for rule *5.2*. Hence, the conclusion of rule *5.2*will be $\{has\_committed(Grace, larceny)\}$ which reads as “Grace has committed larceny”.

6. **CONCLUSION**

This section reports an implementation of a rule learning technique for learning legal rules which are combined with handcrafted legal rules for legal reasoning to answer the US bar examination questions. To summarize, combining our handcrafted rules with the learned rules used for the experimentation with our knowledge base have not succeeded in coming close to getting a reasonable number of the correct answers. The results show poor performance in identifying the correct answers. However, further analysis on the failing cases of the three distributions (A, B, and C), we believe that both the training and test datasets are of insufficient size to fully evaluate the system. In addition, the coverage of the learned legal rules may be

insufficient to perform legal reasoning when answering the bar examination questions. The learned rules cover only murder, larceny, and theft in criminal law as well as search in criminal procedure. We observed that the tool shows promising performance if more questions covering the necessary spectrum were to be used for training the tool. Adding more questions from different criminal law case types (such as kidnapping, vandalism etc) would allow for further evaluation as to whether the system works generally in the criminal law domain. In future more heterogeneous data covering all the spectrum of the law will collected for further experimentation.

REFERENCE

1. Plinere, D., Borisov, A.: SWRL: Rule acquisition using ontology. Scientific Journal of Riga Technical University. Computer Sciences 40 (1), 117 - 122 (2009)

2. Golbreich, C., Imai, A.: Combining SWRL rules and OWL ontologies with Protégé OWL Plugin, Jess, and Racer. 7th International Protégé Conference, Bethesda, MD (2004)

3. Golbreich, C.: Combining rule and ontology reasoners for the semantic web. International Workshop on Rules and Rule Markup Languages for the Semantic Web, pp 6 - 22 Springer (2004)

4. Ruiz-Bertol, F. J., Rodriguez, D., Dolado, J.: Applying rules to an ontology for project management. Proc. of JISBD, pp 5 - 7 (2011)

5. Drabent, W., Henriksson, J., Maluszynski, J.: HD-rules: A Hybrid System Interfacing Prolog with DL-reasoners. ALPSWS 287, (2007)

6. Yang, F., Chen, X.: DL clog: A Hybrid System Integrating Rules and Description Logics with Circumscription. Description Logics (2007)

7. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pp 55 - 60 (2014)

8. Tran, H. D., Stepanova, D., Gad-Elrab, M. H., Lisi, F. A., Weikum, G.: Towards nonmonotonic relational learning from knowledge graphs. International Conference on Inductive Logic Programming, pp 94 - 107. Springer, (2016)

9. Bowman, S. R., Potts, C., Manning, C. D.: Learning distributed word representations for natural logic reasoning. 2015 AAAI Spring Symposium Series (2015)

10. Angeli, G., Premkumar, M. J. J., Manning, C. D.: Leveraging linguistic structure for open domain information extraction. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 1 pp 344 - 354 (2015)

11. Fawei, B., Pan, J. Z., Kollingbaum, M., Wyner, A. Z.: A Methodology for a Criminal Law and Procedure Ontology for Legal Question Answering. Joint International Semantic Technology Conference, pp 198 - 214. Springer, (2018)

12. Corapi, D., Russo, A., Lupu, E.: Inductive logic programming as abductive search. Technical Communications of the 26th International Conference on Logic Programming, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2010)

13. Ray, O.: Nonmonotonic abductive inductive learning. Journal of Applied Logic 7 (3), 329 - 340 (2009)

14. Lisi, F. A.: Inductive logic programming in databases: From Datalog to. Theory and Practice of Logic Programming 10(3), 331 - 359 (2010)

15. Fawei, B., Wyner, A. Z. and Pan, J. Z. Passing a USA National Bar Exam: a First Corpus for Experimentation. In LREC2016, Tenth International Conference on Language Resources and Evaluation. LREC2016.

16. Gad-Elrab, M. H., Stepanova, D., Urbani, J., Weikum, G.: ExFaKT: A Framework for Explaining Facts over Knowledge Graphs and Text. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, ACM pp 87-95 (2019)

17. Lassila, O., Swick, R. R., others.: Resource description framework (RDF) model and syntax specification. Citeseer (1998)

18. Corby, O., Dieng, R., H_ebert, C.: A conceptual graph model for w3c resource description framework. International conference on conceptual structures, pp 468 - 482 Springer, (2000)

19. Lopez, V., Pasin, M., Motta, E.: Aqualog: An ontology-portable question answering system for the semantic web. European Semantic Web Conference, pp 546 – 562 Springer, (2005)

20. Guo, Q., Zhang, M.: Question answering system based on ontology and semantic web. International Conference on Rough Sets and Knowledge Technology, pp 652 – 659 Springer, (2008)

21. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. Proceedings of the IEEE, 104 (1), pp. 11 - 33 (2016)

22. Gal_arraga, L., Teioudi, C., Hose, K., Suchanek, F. M.: Fast rule mining in ontological knowledge bases with AMIE+. The VLDB JournalThe International Journal on Very Large Data Bases, 24 (6) 707 – 730 (2015)

23. Taniar, D., Rahayu, W., Lee, V., Daly, O.: Exception rules in association rule mining. Applied Mathematics and Computation, 205 (2), 735 - 750 (2008)

24. Gad-Elrab, M. H., Stepanova, D., Urbani, J., Weikum, G.: Exception-enriched rule learning from knowledge graphs. International Semantic Web Conference, pp 234 – 251 Springer (2016)

25. Wu, F., Hoffmann, R.,Weld, D. S.: Information extraction from Wikipedia: Moving down the long tail. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge discovery and data mining, pp 731 – 739 ACM (2008)

26. Koller, D., Friedman, N., Dzeroski, S., Sutton, C., McCallum, A., Pfeffer, A., Abbeel, P., Wong, M., Heckerman, D., Meek, C., others.: Introduction to statistical relational learning. MIT press, (2007)

27. Fawei, B., Pan, J. Z., Kollingbaum, M. & Wyner, A. Z. (2019) A Semi-automated Ontology Construction for Legal Question Answering. New Generation Computing, 37(4), pp. 453-478