Automated Qualitative Rule Extraction Based on Bidirectional Long Short-Term Memory Model

Xiaofeng Z.[a], Haijiang L.[a,*], Guanyu X.[a], Honghong S.[b]
[a]Cardiff University, United Kingdom, [b]Dalian Maritime University, China
ZhuX29@cardiff.ac.uk; LiH@cardiff.ac.uk*

Abstract. Digital transformation in the construction industry demands smart compliance checking against relevant standards to ensure high-quality project delivery. Due to the diverse characteristics, the qualitative rule extraction for standards remains labour intensive. Therefore, an efficient and automated rule extraction method is pivotal. The artificial neural network has been widely used for textual feature extraction in recent years. In this paper, the authors construct an automated rule extractor based on a bidirectional Long short-term memory (LSTM) neural network model, which can automate the extraction of qualitative rules in textual standards and achieves an accuracy of 96.5% in actual tests. The automated rule extractor can greatly improve the efficiency of converting unstructured textual rules to structured data. This approach can establish the basis for knowledge mining of qualitative standards as well as the development of large-scale compliance checking systems.

1. Introduction

Research on the automated extraction of rules from standards has been carried out since the 1960s (Xue and Zhang, 2021). Rules are accepted principles or instructions that state the way things are or should be done, which normally can be classified as quantitative or qualitative according to their quantifiability (Marsal-Llacuna, 2018). Taking the provisions of IBC 2015 in Table 1 as an example, quantitative rules in standards are usually expressed directly through specific numerical values with units of measurement, which can be easily recognised by regular expression matching. As it is challenging to quantify the qualitative rules (provision 2 in Table 1), the requirements are generally described in a textual manner. This purely textual description is distinguished by its variety of forms, flexibility of structure, and logical complexity, which increases the difficulties in rule localization and feature identification when extracting qualitative rules automatically (ul Hassan and Le, 2020).

Table 1: Three examples of the IBC 2015 provisions

Number	Description	Type
1	The fire separation distance between a building with polypropylene siding and the adjacent building shall be not less than 10 feet (3048 mm) (Clause 1402.12.2)	Quantitative rule
2	Exterior walls, and the associated openings, shall be designed and constructed to resist safely the superimposed loads. (Clause 1403.3)	Qualitative rule
3	The purpose of this code is to establish the minimum requirements to provide a reasonable level of safety, public health, and general welfare through structural strength… (Clause 101.3)	Explanation of rules

For the engineering domain, several rule-based and machine learning extraction algorithms have been proposed in academia (Gunjan and Bhattacharyya, 2021). Compared to the rule-based methods which are inflexible, under-generalized and time-consuming, machine learning-based algorithms are generally more efficient but less accurate (Hailesilassie, 2016). Thus, the dominant rule extraction method in the engineering domain is still manual. With the continuous development and improvement of standards, manual extraction gradually falls short in two scenarios: multi-standards comparison and administration of complex engineering projects such as quality assurance and process management. On other hand, the rule-based and the machine learning-based approaches cannot meet the demands of the industry either, due to their incapability in efficiency and accuracy.

The authors have therefore developed a rule extractor based on a bi-directional LSTM model that enables fast, accurate and efficient automated mining of qualitative rules. To develop the rule extractor, the authors manually created a textual dataset of rules and enhanced it with data augmentation methods to fulfill a uniform sample distribution and then further improved the reliability and objectivity of the dataset through the Delphi method. After that, this dataset was then trained to produce word vectors representing the probability distributions of their correspondence in the engineering domain and adopted in the training of a bidirectional LSTM neural network to form the rule extractor. The trained rule extractor eventually achieved 98% accuracy on the test set and 96.5% accuracy in an actual validation of a qualitative standard.

An automated rule extractor can significantly reduce the proportion of manual work involved in rule extraction and process a large number of texts of standards in a short period. So, the arises of an automated rule extractor make it possible to check compliance among multiple qualitative standards and to conduct large-scale knowledge mining for qualitative standards in the engineering domain.

2. Related Work

2.1 Overview of Rule Extraction Algorithms

The keyword method is considered the most basic rule extraction method by identifying target vocabularies (Bednár, 2017), examples are the IF-THEN rule and the M-of-N rule. An improved approach is to leverage libraries of domain-specific languages, especially knowledge from domain experts. However, the accuracy of the keyword method lacks generalisability and highly relies on the libraries provided (Dragoni et al. 2016).

With the development of machine learning, some more intelligent techniques for rule extraction have been developed. Machine learning-based methods can be classified into heuristic approaches, regression-based approaches and artificial neural network approaches (Gunjan and Bhattacharyya, 2021). In heuristics, decision tree ensembles (DTEs) such as random forest (RF) give high prediction accuracy while being regarded as black-box models (Abellán et al., 2018). Support vector machine (SVM), as one of the regression-based approaches, exhibits good prediction performance when applied to several publicly available data. The downside is that it requires a pre-defined training model and falls short of scalability (Barakat and Bradley, 2010).

Artificial neural networks (ANN) are more advanced machine learning methodologies for rule extraction which can be categorised into decompositional, pedagogical and eclectic approaches, as proposed by Andrews et al. (1995). The decompositional method works by synthesising the activation rules, along with weights and biases of the hidden layers of the neural network. The pedagogical rule extraction works by matching the input-output relationship to the way that neural networks interpret it (Gethsiyal Augasta and Kathirvalavakumar, 2012). While the eclectic approach is a hybrid between pedagogical and decompositional methods, it generally analysis the ANN at the individual unit level but extracts rules at the global level (Hruschka and Ebecken, 2006). A decompositional technique is substantially more translucent compared with pedagogical algorithms, but it can be time-consuming as it is delivered layer by layer (Zarlenga et al., 2021). While in terms of computational limitation and execution time, the pedagogical approach usually delivers more performance than the decompositional.

2.2 Delphi Method

The Delphi method is a structured decision support technique that aims to obtain relatively objective information, opinions and insights through the independent and iterative subjective judgment of multiple experts in the information collection process (Hsu and Sandford, 2007). The survey team conducted multiple rounds of consultation with selected expert groups through anonymous approaches and then compiled the expert opinions from each round. This process is repeated several times until there is a convergence of opinion, resulting in a more consistent and reliable conclusion or solution.

2.3 Data Augmentation

Data augmentation techniques were originally applied in the field of computer vision, where the core approach was to create new image data by panning, rotating, compressing and adjusting the colour of an image (Xie et al., 2022). Unlike image data, data in natural language is discrete. Therefore, it is not feasible to perform a direct and simple transformation of the input data, which would most likely change the original meaning of the sentence. Currently, the two categories of text data augmentation methods commonly used in natural language processing are text representation-oriented augmentation and text-oriented augmentation. Text representation-oriented augmentation focuses on the processing of the feature representation of the original text, while original text-oriented augmentation is carried out by replacing synonyms or deleting words in the original text (Shorten et al., 2021). Back translation is a typical method of text-oriented augmentation, which re-translates content from the target language back to its source language in literal terms to obtain a sentence with a similar meaning but in a different expression (Edunov et al., 2018). This method enables not only the substitution of synonyms and the addition or deletion of words but also the reconstruction of the word order in sentences. Thus, back-translation is a very effective and reliable approach to augmenting textual data.

2.4 LSTM Neural Network

Long short-term memory (LSTM) neural networks are a special type of recurrent neural network (RNN) model, which is widely utilized in natural language processing and time-series prediction (Hochreiter and Schmidhuber, 1997). In contrast to RNNs, LSTMs introduce the concept of cell state and control the update of information on cell states through gate states, thus solving the problem of long-term dependencies. As a consequence, LSTM models can acquire better performance on feature learning of long sequences. In natural language, the meaning of a word or sentence is not only influenced by the previous content but is also related to the following content (Schuster and Paliwal, 1997). A bidirectional LSTM model that learns features from both forward and backward directions can capture more information and obtain better training results.

3. Methodology

The automated extractor of the qualitative rule is essentially a binary classifier trained from a neural network model. The underlying principle of qualitative rule extraction is similar to that of sentiment classification in natural language processing, in that the neural network model is trained with a large number of annotated rule texts to learn the sequential and lexical features of the textual rules, thus achieving the identification and extraction of qualitative rules.

Therefore, this research followed the training process of neural network models, including six main steps: dataset creation, data pre-processing, word vector embedding, neural network model construction, data loading, training and validation. The specific flowchart of this research is shown in Figure 1.

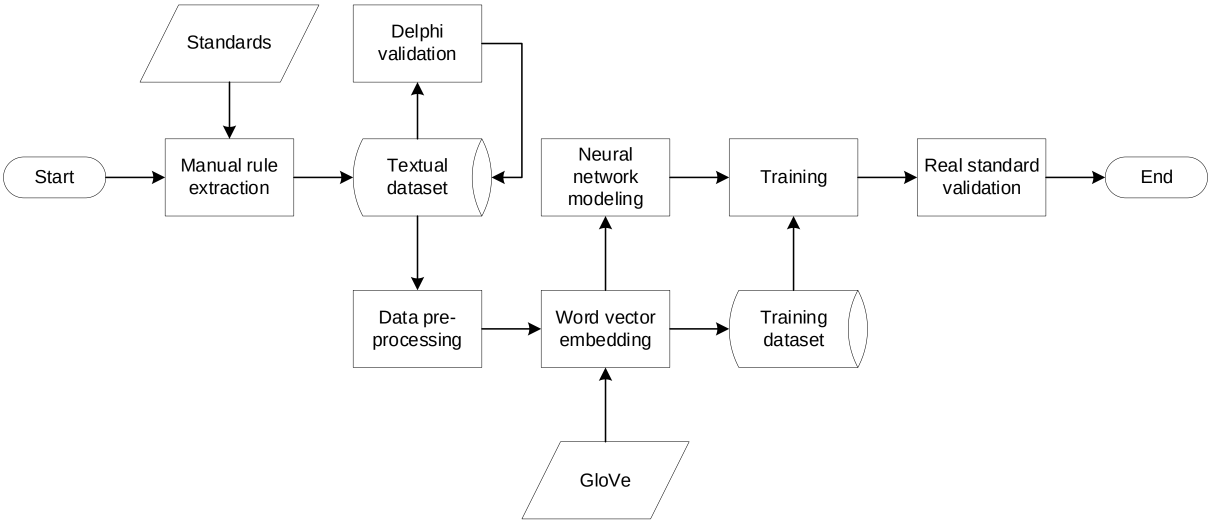

Figure 1: The development flowchart for an automated qualitative rule extractor

As all the qualitative rules in the dataset were manually extracted from the standard files, the labels of the rule samples in the dataset are inevitably subjective. In this research, the authors adopted the Delphi method to validate the labels of the rule samples to improve the objectivity and reliability of the dataset. To address the problem of uneven data distribution, the authors applied the Back Translation method to optimise the dataset and further improve the robustness and generalisation of the trained model. Considering that most of the qualitative rules in the dataset are derived from engineering standards, the authors embedded words in the dataset with word vectors trained from the lexicon of the engineering domain to achieve higher accuracy. At the end of this research, the trained model was validated using a specific engineering standard to investigate the actual performance of the extractor.

Due to space limitations, this paper only focuses on solving the problem of locating and identifying qualitative rules in the standard. The extracted rules still retain the original sentence format, which is the same as the unextracted rules in Table 3. The extraction of entities and relationships within the qualitative rules and the transformation to computer-processable rules will be further illustrated in the following papers.

4. Case Study

4.1 Text Dataset Creation

For the qualitative rule extractor in this research, there is no open dataset that fully satisfies the experimental requirements. The author, therefore, chose to manually build a small sample dataset and combine it with textual data augmentation methods to develop a dataset that is suitable for training. To ensure that the extractor not only can solve the research problem but also has sufficient generalisation capabilities, the research group manually extracted 826 rule samples from 6 international and national engineering standards, which are listed in Table 2. These standards cover a wide range of aspects of the building and construction domain, including quality management, energy management, information management process, building design, etc.

Table 2: Engineering standards included in the rule dataset.

Standard Code	Description	Published by	No. of Rules extracted
ISO 9001	Quality management systems Requirements	ISO[a]	216
ISO 14001	Environmental management systems -Requirements with guidance for use	ISO	169
ISO 50001	Energy management systems - Requirements with guidance for use	ISO	153
ISO 19650-1	Organization and digitization of information about buildings and civil engineering works, including building information modeling (BIM)	ISO	199
2015 IBC	International Building Code	ICC[b]	48
GB/T 51212	Unified standard for building information modeling	MOHURD[c]	41

[a]International Organization for Standardization, [b]ICC - International Code Council, [c]MOHURD – Ministry of Housing and Urban-Rural Development of China

4.2 Data Augmentation

After completing the initial extraction of the qualitative rules, there were 826 samples in the dataset, containing 573 rule samples and 253 non-rule samples. In general, the number of rule texts in the standard file is more than the number of non-rule texts. Currently, it is generally accepted that 1000 samples per category are required to be collected to achieve more accurate results when training neural network models for classification. In other words, there should be 2000 samples in the dataset of this research, including 1000 qualitative rule samples and 1000 non-rule samples. To meet the requirement of sample size, the authors chose to utilize back translation in data augmentation techniques to expand the dataset and balance the number of positive and negative samples. In the experimental process, the research group selected the more commonly used French and Mandarin as intermediate languages to implement the back translation of the samples. The specific procedure was as follows.

1) Count the number of positive and negative samples to be augmented
2) Randomly select a corresponding number of positive and negative samples in the dataset according to the statistical results
3) The selected samples were translated into French and Mandarin one by one with the help of a third-party translator (Google translation), and then translated back into English by another translator (DeepL) to become the translated samples.
4) The newly translated samples were compared with the original samples, and if their expressions were different, the translated samples were added to the dataset as augmented samples.

The augmented rule dataset contains 2000 samples, of which 1000 are positive rule samples and 1000 are negative non-rule samples. The dataset developed in this research can be accessed through https://gitlab.com/LonelyRanger/rules-extraction.git

4.3 Delphi Dataset Validation

The problem of small sample size and uneven sample distribution in the dataset has been addressed utilizing data augmentation. Another issue that needs to be addressed is the reliability of the samples and labels in the dataset. In this research, the research team followed the Delphi method to bring together five experts with a deep understanding of engineering standards to validate the samples and labels in the dataset. The validation process was as follows.

1) The full sample and label data underwent the first round of expert group validation.
2) The research group collates and tallies the validation results from the expert group. For samples where all experts agree, they can pass the validation directly. If more than half of the experts agree on the sample, the research group modifies the sample appropriately based on the experts' opinions and then validates it in the next round. If the sample passes the validation by only a few experts, a new sample of the same label type is generated by data augmentation to replace the original sample and is validated in the next round.
3) All adjusted samples were subjected to the above two steps of validation again until all experts came to an agreement.

In this research, after completing four rounds of validation and modification, the five experts reach an agreement on the qualitative rules and labels for the dataset. At this point, the 2000 samples in the dataset possessed a high degree of objectivity and reliability.

4.4 Textual Data Pre-processing

In this research, the pre-processing of the textual data is divided into four steps: (1) removing non-textual parts of the samples, such as punctuation, special characters, etc.; (2) splitting the rule samples into sequences consisting of individual words using the word splitting function; (3) removing stop words from the individual sample sequences. (4) counting the characteristics of the data set, such as the number of samples, the maximum sequence length, the size of the lexicon, etc. The rule samples in the dataset are all derived from standard clauses, which have a normative presentation. Therefore, noise removal and text normalisation are not necessary for the pre-processing stage of this research.

4.5 Word Embedding

Text is a piece of unstructured data information that cannot be directly computed. Word embedding is a digital text representation method that transforms unstructured textual data into structured data that can be recognised and computed. The more mainstream methods currently utilize word vectors (Word2vec, GloVe (Pennington et al., 2014)) pre-trained on large corpora as word representations. This research is aimed at the engineering domain, where the probability distribution of words is somewhat different from that of the larger corpus. The pre-trained GloVe word vector is adopted as the initial value for word embedding and modified by the backpropagation algorithm during the training process according to the distribution pattern of the vocabulary in the dataset, which can make it more consistent with the probability distribution of the vocabulary in the engineering domain.

4.6 Neural Network Training

According to the statistics of the text pre-processing stage, the maximum length of the sample sequences in the dataset was 52 and the average length was 23.56. Considering the advantages of LSTM models in long sequence learning, the authors selected a bidirectional LSTM model as the training model for the rule extractor. The bidirectional LSTM model in this research is composed of one embedding layer, two LSTM layers and one Dense layer, containing a total of 303,775 trainable parameters. The model structure is shown in Figure 2. Each LSTM layer is followed by a dropout layer with dropout rates of 0.5 and 0.25, respectively.

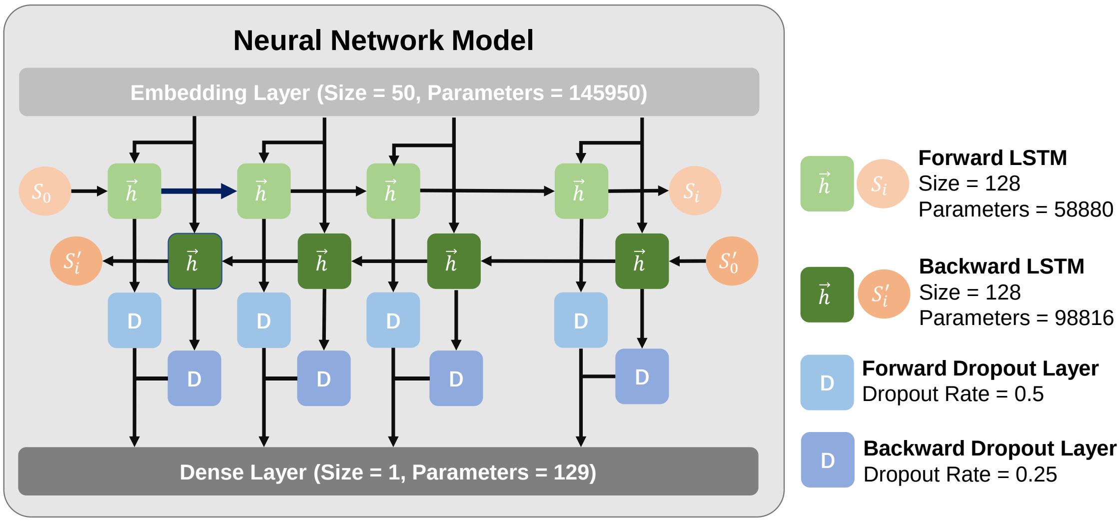

Figure 2: Structure of the neural network model for automated qualitative rule extraction

For the partitioning of the dataset, the authors applied random sampling to divide the entire dataset into a training set, a validation set and a test set in the ratio of 6:2:2. The hyperparameters of the neural network model, such as batch size, trainable layer size and dropout rate, are set based on the authors' previous experimental experience in neural network model training. In addition, neural network models with these hyperparameters are widely used in various natural language processing tasks and usually perform well in learning. Another hyperparameter, epoch, cannot be predicted before training. Given the low diversity of the dataset, the authors determined to train the model with 10 epochs, 20 epochs and 30 epochs to investigate the detailed training performance of the model.

4.7 Actual Standards Validation

The dataset in this research was built manually by the research group. Hence, the textual feature distribution may slightly differ from the real standard file, which is possibly lead to a deviation in the actual performance of the rule extractor. Therefore, the generalisation error of the rule extractor is required to be verified through actual tests after the model has been trained. The authors use the trained extractor to extract qualitative rules from a new engineering standard (ISO19650-2) and compare the automatic extraction results with those of the manual extraction by experts to evaluate the performance of the rule extractor in practice.

5. Result

To find a better performing training epoch, the authors trained the model for 10 epochs, 20 epochs and 30 epochs, and observed the changes in loss and accuracy during training. As shown in Figure 3, after 10 epochs of training, the loss of the model underwent a continuous decline, which indicates that the model did not reach the optimal training state. After 20 epochs of training, the loss of the model is still decreasing but gradually stabilizing, suggesting that the model is close to the optimal state. After 30 training epochs, the model reached its optimal state, with both loss and accuracy fluctuating within a small range. The trained model achieved a final accuracy of 98% on the test set.

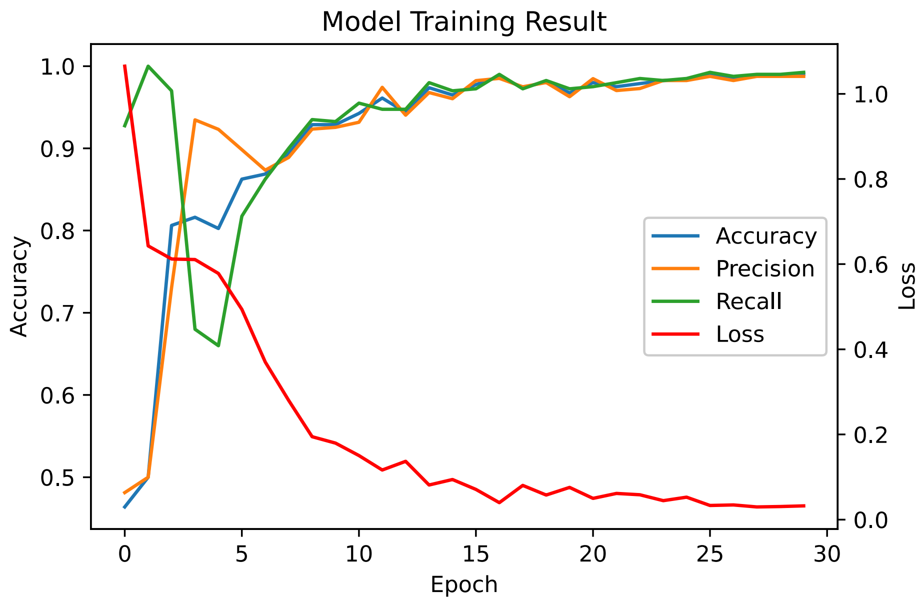

Figure 3: Training results under different training epochs

In the session of the actual standards validation, the expert group manually extracted a total of 228 qualitative rules from ISO19650-2. Relatively, the trained rule extractor extracted 220 rules from ISO19650, which achieved an accuracy rate of 96.5%. The eight missing qualitative rules are listed in Table 3. The rule extractor can therefore be tentatively concluded to have a good performance in the qualitative rule extraction for engineering standards.

Table 3: List of the 8 unextracted rules in the actual standard validation.

Number	Description of Unextracted Rules
1	The national standards organizations of the following countries are bound to implement this European Standard.
2	Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee.
3	ISO shall not be held responsible for identifying any or all such patent rights.
4	This document is applicable to built assets and construction projects of all sizes and all levels of complexity.

5	The amount of thought involved, the time taken to complete it and the need for supporting evidence will depend on the complexity of the project.
6	This document can be used by any appointing party.
7	The appointing party's defined information exchange points within each of the principal work stages are to be used in defining the project's information requirements.
8	It is recommended that this is done as a separate appointment before procurement of any other appointed party starts.

6. Discussion

The diverse features of qualitative rules make it a challenge for traditional algorithms to identify and locate rules. The manual extraction of rules is essentially a way of summarising the features of the text in the standards through human comprehension, thus achieving accurate recognition of the rules. This style of learning is mechanistically identical to the training of artificial neural networks that mimic the human brain's nervous system. Therefore, theoretically it is possible to enable automated rule extraction by training artificial neural networks. Compared with an open language environment, the expression and description of the rules in the standards are more normative, which implies the textual features are more distinct. Consequently, the trained neural network model was able to acquire a high accuracy on the test set.

In addition to the training results of the model itself, the actual performance of the rule extractor is also closely related to the quality of the dataset. Only if the dataset has a similar sample distribution to that of the application environment can the extractor acquire an accuracy close to the test set in actual applications. Through this research, the authors provide researchers who are facing similar problems with a solution to quickly build reliable small sample domain datasets.

In the actual standard validation, the rule extractor achieves an accuracy of 96%, which indicates that the rule extractor can automate rule extraction to some extent, and significantly reduce the manual work involved. However, given the nature of the rule extraction task, the extractor can only fully substitute manual extraction if its extraction accuracy is close to 100%. As can be seen from Table 3, the rule extractor has some deficiencies in the recognition of rules with a weak feature. This may be caused by the non-linearity of the activation function. In this research, the Softmax function is utilised as the activation function for output, which may be not sensitive enough for a weak feature. The research group will try some other activation functions to improve the recognition performance in future research. In addition, the use of abbreviations (unextracted rule 3) results in the loss of semantic information in the context, which may affect the recognition of qualitative rules to some extent.

In the experimental process of this study, the selection of the neural network model and the setting of some of the hyperparameters are based on the authors' previous research experience. Consequently, further research is needed on model selection and the setting of hyperparameters to obtain the best training performance. In addition, there is also room for further reduction of the generalisation error of the rule dataset.

Although this research focuses on qualitative rules in the engineering domain, the authors believe that this rule extractor is essentially capable of extracting rules from other domains due to their similarities in textual characteristics. For a more accurate result, transfer learning can be applied with this rule extractor as a pre-training model.

7. Conclusion

Automated extraction of qualitative rules is a pressing challenge in the research of engineering. In this research, the authors built a highly reliable domain dataset employing data augmentation and Delphi expert validation, and further developed an automated qualitative rule extractor by training a bi-directional LSTM model on this dataset. The extractor achieves an accuracy of 96.5% on the actual standards test and is therefore primed for application to practical tasks. For future work, alternative artificial neural network models and hyperparameters can be investigated for potential improvement in accuracy. It is expected that the automated rule extractors can replace traditional manual extraction methods to a certain extent and largely improve the efficiency of transforming unstructured textual rules into structured rules. The realisation of automated extraction of qualitative rules is of great significance for the future establishment of large-scale semantic rule knowledge bases and multi-standard compliance checking systems.

References

Abellán, J., Mantas, C.J., Castellano, J.G. and Moral-García, S. (2018). Increasing diversity in random forest learning algorithm via imprecise probabilities. Expert Systems with Applications 97, pp. 228–243. doi: 10.1016/J.ESWA.2017.12.029.

Andrews, R., Diederich, J. and Tickle, A.B. (1995). Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge-Based Systems 8(6), pp. 373–389. doi: 10.1016/0950-7051(96)81920-4.

Barakat, N. and Bradley, A.P. (2010). Rule extraction from support vector machines: A review. Neurocomputing 74(1–3), pp. 178–190. doi: 10.1016/J.NEUCOM.2010.02.016.

Bednár, P. (2017). Vocabulary matching for information extraction language. In: 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)., pp. 149–152. doi: 10.1109/SAMI.2017.7880292.

Dragoni, M., Villata, S., Rizzi, W. and Governatori, G. (2016). Combining Natural Language Processing Approaches for Rule Extraction from Legal Documents. In: AI Approaches to the Complexity of Legal Systems., pp. 287–300. Available at: https://wordnet.princeton.edu/ [Accessed: 25 February 2022].

Edunov, S., Ott, M., Auli, M. and Grangier, D. (2018). Understanding Back-Translation at Scale. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018. Association for Computational Linguistics, pp. 489–500. Available at: https://arxiv.org/abs/1808.09381v2 [Accessed: 15 February 2022].

Gethsiyal Augasta, M. and Kathirvalavakumar, T. (2012). Rule extraction from neural networks; A comparative study. In: International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012)., pp. 404–408. doi: 10.1109/ICPRIME.2012.6208380.

Gunjan, A. and Bhattacharyya, S. (2021). A Brief Review of Intelligent Rule Extraction Techniques. Advances in Intelligent Systems and Computing 1333 AISC, pp. 115–122. doi: 10.1007/978-981-33-6966-5_12.

Hailesilassie, T. (2016). Rule Extraction Algorithm for Deep Neural Networks: A Review. IJCSIS) International Journal of Computer Science and Information Security 14(7), pp. 376–381. Available at: https://sites.google.com/site/ijcsis/ [Accessed: 23 February 2022].

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation 9(8), pp. 1735–1780. Available at: https://direct.mit.edu/neco/article/9/8/1735/6109/Long-Short-Term-Memory [Accessed: 23 February 2022].

Hruschka, E.R. and Ebecken, N.F.F. (2006). Extracting rules from multilayer perceptrons in classification problems: A clustering-based approach. Neurocomputing 70(1–3), pp. 384–397. doi: 10.1016/J.NEUCOM.2005.12.127.

Hsu, C.-C. and Sandford, B. (2007). The Delphi Technique: Making Sense of Consensus. Practical Assessment, Research, and Evaluation 12. doi: 10.7275/PDZ9-TH90.

Marsal-Llacuna, M.L. (2018). The standards evolution: A pioneering Meta-standard framework architecture as a novel self-conformity assessment and learning tool. Computer Standards & Interfaces 55, pp. 106–115. doi: 10.1016/J.CSI.2017.06.002.

Pennington, J., Socher, R. and Manning, C.D. (2014). GloVe: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)., pp. 1532–1543. Available at: http://nlp. [Accessed: 16 February 2022].

Schuster, M. and Paliwal, K.K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45(11), pp. 2673–2681. doi: 10.1109/78.650093.

Shorten, C., Khoshgoftaar, T.M. and Furht, B. (2021). Text Data Augmentation for Deep Learning. Journal of Big Data 8(1). Available at: https://doi.org/10.1186/s40537-021-00492-0.

ul Hassan, F. and Le, T. (2020). Automated Requirements Identification from Construction Contract Documents Using Natural Language Processing. Journal of Legal Affairs and Dispute Resolution in Engineering and Construction 12(2). Available at: https://orcid.org [Accessed: 23 February 2022].

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T. and Le, Q. V. (2022). Unsupervised Data Augmentation for Consistency Training. In: 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Available at: https://github.com/google-research/uda. [Accessed: 14 February 2022].

Xue, X. and Zhang, J. (2021). Part-of-speech tagging of building codes empowered by deep learning and transformational rules. Advanced Engineering Informatics 47. doi: 10.1016/J.AEI.2020.101235.

Zarlenga, M.E., Shams, Z. and Jamnik, M. (2021). Efficient Decompositional Rule Extraction for Deep Neural Networks. arXiv e-prints , p. arXiv:2111.12628. Available at: https://github.com/mateoespinosa/remix.