Detecting Regulation Violations for an Indian Regulatory Body through Multi Label Classification

Ujwal Narayan
ujwal.narayan@research.iiit.ac.in
International Institute of Information Technology
Hyderabad, India

Pulkit Parikh
pulkit.parikh@research.iiit.ac.in
International Institute of Information Technology
Hyderabad, India

Kamalakar Karlapalem
kamal@iiit.ac.in
International Institute of Information Technology
Hyderabad, India

Natraj Raman
natraj.raman@jpmorgan.com
JPMorgan AI Research
London, UK

ABSTRACT

The Securities and Exchange Board of India (SEBI) is the regulatory body for securities and commodities in India. SEBI creates, and enforces regulations that must be followed by all listed companies. To the best of our knowledge, this is the first work on identifying the regulation(s) that a SEBI-related case violates, which could be of substantial value to companies, lawyers, and other stakeholders in the regulatory process. We create a dataset for this task by automatically extracting violations from publicly available case-files. Using this data, we explore various multi-label text classification methods to determine the potentially multiple regulations violated by (the facts of) a case. Our experiments demonstrate the importance of employing contextual text representations to understand complex financial and legal concepts. We also highlight the challenges that must be addressed to develop a fully functional system in the real-world.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Information systems** → *Clustering and classification*; • **Applied computing** → **Law**.

KEYWORDS

regulation violation detection, BERT, multi label classification, information extraction, neural networks

ACM Reference Format:
Ujwal Narayan, Pulkit Parikh, Kamalakar Karlapalem, and Natraj Raman . 2022. Detecting Regulation Violations for an Indian Regulatory Body through Multi Label Classification. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3487553.3524640

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524640

1 INTRODUCTION

The Securities and Exchange Board of India (SEBI) is the regulatory body for securities and commodity market in India. SEBI ensures that the interest of the investors in securities are protected, and promotes the development of as well as the regulation of the securities market. They are both a quasi-legislative body passing regulations and acts that companies must abide, as well as a quasi executory and judicial body where they ensure that any non-compliance is identified and the relevant penalties are applied. SEBI mandates that every company must have a compliance office who is in charge of ensuring that the regulations are followed.

While there are many regulations that companies must abide, only a few of those regulations such as the prohibition of insider trading are frequently violated. Violations of the same regulations have similar sets of circumstances, events and actions which together form the facts of the case. Thus it is possible to leverage these facts to detect if a regulation has been violated. Doing so could aid only the compliance offers of companies but also other actors in the regulatory process such as lawyers. Towards this, we design and develop a method to automatically detect the regulation violation, and if a violation has been detected, we identify the one or more rules or regulations it violates.

We leverage SEBI case-files and regulations to develop a large corpora of regulation violations that we use to train and validate our classification approaches. We depict our pipeline in Fig: 1. We compare with multiple classification models and conclude that using context-aware language models is critical to understanding the complex linguistic patterns present in regulatory documents. As general purpose language technologies are not well suited for these domain specific tasks, we fine tuned BERT in the SEBI domain, and then leverage this SEBI-BERT to perform multi label classification.

Furthermore, we discover that it is necessary to understand the facts of a case in entirety and account for rippling change effects. We believe that our findings would aid in better understanding of the regulatory documents. Our code and data are publicly available on GitHub[1]

[1]https://github.com/JPMS-DSAC/regulation_violation_detection

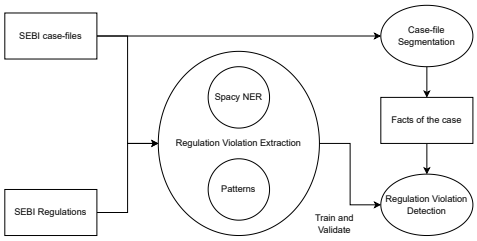

Figure 1: Pipeline of the regulation violation detection system

Table 1: Dataset statistics

Data	Number of Files
SEBI Regulations	76
SEBI Case-files	8500

2 DATA

2.1 SEBI Regulations

SEBI publishes the list of rules and regulations that all companies must abide by, and periodically updates these rules and regulations. The latest versions of these regulations are scraped from the SEBI web-page [2]. The regulations obtained in PDF format are converted to text for further processing. The regulations are organised into sections, and each regulation can have various sub-regulations. We create a unique mapping for each regulation based on file name, section, regulation and sub-regulation number to the regulation text, and obtain 750 such regulations.

2.2 SEBI Case-files

Appeals against the SEBI orders are handled in the Securities Appellate Tribunal (SAT). The judgements from these appeals are publicly available at the SEBI web page[3]. We scrape this page, and obtain $\tilde{8}500$ judgement orders. Since these judgements are available in PDF formats, we convert them to text, and then clean it up of formatting and rich data such as tables or images if any.

2.3 Regulation Violation Extraction

We automatically extract the regulation violations in an unsupervised manner from the given case files. We do this by leveraging the regulation information. We use Spacy's[4] Named Entity Recognition (NER) model to detect the mention of a regulation or law in the SEBI case-files. As the publicly available model is not suitable for SEBI data, we extend it by annotating fifteen case-files. We evaluate the model on a held out corpus of ten case-files.

As the number of ways in which the regulation violation is reported is limited, we construct rules to match these patterns. **Example:** A regulation is said to be violated if particular block of text in the case-file matches the list of rules, has the "LAW" tag from the fine tuned spacy NER, and the 'LAW' entity belongs to the collected list of regulations.

While there are 750 different regulations, most of the regulations have minimal violations reported. A small set of regulations contributed to the majority of the violations. By grouping the least frequently violated regulations into a single category, "rest", we use a set of 50 classes, where each category has at least 50 reported violations. Table 2 shows the most frequently violated regulations.

Thus broadly speaking, the dataset we collect can be divided into two parts:

- SEBI Regulations:
  These are the set of acts, rules and regulations that govern the actions and interactions of companies, and individuals such as promoters, insiders etc.
- SEBI Case-files:
  When a particular rule or regulation is deemed to be violated, SEBI launches legal actions against the relevant parties. The parties have a right to appeal SEBI's decisions and the proceedings of these appeals are termed as casefiles.

Table 1 shows the statistics for the different parts.

Table 2: Regulation IDs with their Frequency of violation in the corpus.

Regulation ID	Frequency of Violation	Regulation ID	Frequency of Violation
sectionSEBI_15j	4414	sast_13(1)	125
sectionSEBI_15h	4031	sast_7(2)	92
sectionSEBI_4(1)	238	sectionSEBI_15a(a)	242
sectionSEBI_15a	2370	sast_8(2)	68
pit_13(1)	333	sectionSEBI_3(a)	130
pit_13(3)	339	sectionSEBI_4(2)	201
takeover_13(1)	80	sectionSEBI_12a(a)	56
sectionSEBI_15i	833	sectionSEBI_11(4)	59
sectionSEBI_15c	446	sectionSEBI_11(1)	55
pfutp_4(1)	400	sast_29(1)	85
pfutp_3(a)	302	sast_29(3)	118
pfutp_4(2)	676	sast_8(3)	87
pit_3(i)	92	pit_12(2)	75
pit_12(1)	92	sast_13(3)	90
sectionSEBI_15g	158	sast_29(2)	116
sast_11(1)	57	sast_13(5)	64
pit_13(6)	143	sectionSEBI_15f	69
pit_13(4a)	219	sectionSEBI_12(1)	52
pit_13(4)	245	sectionSEBI_11c(2)	102
pit_7(1)	121	sectionSEBI_11c(3)	64
pit_13(5)	471	sectionSCR_2(i)	86
takeover_7(1)	97	sast_7(1a)	92
sectionSEBI_15a(b)	374	sectionSEBI_5(2)	54
sast_7(1)	129	sectionSEBI_13(5)	55
sectionSEBI_(3)	57	pfutp_3(d)	52

Regulations are abbreviated as follows:

- sectionSEBI: SEBI Act
- pit: Prohibition of Insider Trading Act
- takeover: Substantial Acquisitions of Shares and Takeovers 1997
- sast: Substantial Acquisitions of Shares and Takeovers 2011,
- futp: Prohibition of Unfair Trade Practices

3 METHODOLOGY

We pose the problem of regulation violation detection as a multi label classification task. Given the facts of the case and r different

[2]https://www.sebi.gov.in/legal.html
[3]https://www.sebi.gov.in/sebiweb/home/HomeAction.do?doListing=yes&sid=2&ssid=9&smid=1
[4]https://spacy.io/

regulations, the classification model predicts a one-hot vector of length R, where dimension r being 1 implies that regulation r is violated.

3.1 Facts of the case

As we are interested in detecting regulation violation given a set of facts of the case, we develop a semantic segmentation engine to separate out the different sections of the case-file. With the help of our legal experts, we create a set of ten labels that apply to the sentences of the case files. We collectively refer to sentences that belong to one of the four labels given below as facts of the case.

- **Statutory Facts**: Statements that invoke rules, regulations, acts and orders by the SEBI, either by using their representative names and numbers or by quoting them in totality.
  Example: *Regulation 4(1) of PFUTP Regulations prohibits persons from indulging in a fraudulent or an unfair trade practice in securities.*
- **Procedural Facts**: Statements that contain generic information on the procedure duly followed by the authorities to set the process of adjudication in motion
  Example: *SEBI conducted an investigation in respect of buying, selling and dealing in the shares of GCL during the time period from September 01, 2004 to November 05, 2004 (hereinafter referred to as investigation period).*
- **Material Facts**: Statements that contain information about the case that is relevant and important in deciding the outcome as well as the violation and penalty, if any.
  Example: *The price reached the period low (intra day) of Rs. 1.16 on October 26 & 27, 2004 and finally closed at Rs. 1.53 on November 05, 2004.*
- **Related Facts**: Statements made in a general sense, including truisms, re-emphasis of statutory facts which do not constitute the facts of the instant case, but are material in deciding its outcome.
  Example: *The Hon'ble Supreme Court of India in the matter of SEBI Vs. Shri Ram Mutual Fund [2006] 68 SCL 216 (SC) interalia held that "once the violation of statutory regulations is established, imposition of penalty becomes sine qua No of violation and the intention of parties committing such violation becomes totally irrelevant.*

Twenty seven case files were annotated by our legal experts and we use this data to train a sentence classification model. We formulate a context aware text classification architecture that makes use of the uncased BERT [4] model. In addition to the target sentence, the adjacent left and right sentences in the document are used as model inputs to better capture long range dependencies. After tokenization, they are passed through a BERT layer to get word-level embeddings. The CLS token embeddings for the three sentences are concatenated and the resultant vector is passed through two dense layers for obtaining the desired output label. We train the model with an 80:20 split for training and testing and obtain an F1 score of 0.75

3.2 Machine Learning Approaches

We experiment with several different machine learning classification set ups. We construct baselines through traditional machine learning classifiers with different sets of features. We also experiment with more complex neural architectures such as LSTMs and Transformers.

3.2.1 Traditional Machine Learning. We experiment with Support Vector Machines [3], Random Forests (RF) [6] and Multi Layer Perceptrons (MLP) [5]. In SVMs, we experiment with both linear and kernel SVMs (specifically the Radial Basis Function (RBF)). For features, we explore TF-IDF[10] both at the character level (1-5 character n-grams) and world level (unigram and bigram). We also experiment with generating case-level embeddings with the mean of the sentence embeddings. These sentence embeddings are generated by computing the mean of the GloVe[9] representation for all the words in the sentence.

3.2.2 LSTM based approaches. In addition to the standard LSTM [7] text classification problem formulation, we also experiment with hierarchical LSTMs where the words in the sentence are fed into an LSTM to compute sentence embeddings. These sentence embeddings in turn are fed into another LSTM to compute case level embeddings which is then used to detect the regulation violations. [11]

3.2.3 Transformer based approaches. The recent years have shown how effective transformer based architectures are for various NLP tasks, and thus we also develop a transformer based multi label classifier.

However, these models perform poorly on domain specific tasks such as those in the scientific, medical or legal domain. A possible solution in these closed domains is to fine-tune the model on domain specific data, and approaches such as Sci-BERT [1], BioBERT[8] have shown that fine-tuning can greatly improve performance in these closed domains. Thus, we fine-tune BERT for the SEBI domain to create "SEBI-BERT". We fine tune on a corpus consisting of SEBI regulations, SEBI case-files as well as a collection of financial and SEBI related news articles.

The facts of the case can be large, and thus will not fit in the BERT context span. To solve this, we use the sliding window technique. Any sequence greater than the maximum token length is split into multiple windows. In order to limit the loss of information with hard cutoffs, we ensure that there exists an overlap of at least 20% of the tokens between any two contiguous windows. When predicting, we predict on each of the sub-windows individually, and the final output is aggregated through majority voting.

4 RESULTS

For all the classification experiments we split the data in a $80:20$ ratio as shown in table 3 and perform K fold cross validation with $K = 10$.

We evaluate our classification results with two metrics namely Hamming Loss and F1 Scores.[2]. High F1 scores and low hamming loss values indicate better classifier performance.

Table 3: Data Split for training and testing.

Data	Number of Files
Training	6800
Testing	1700

- Hamming loss (HL) is defined as the fraction of the labels incorrectly predicted and is given by

$$HL = \frac{1}{NL}\sum_{l=1}^{L}\sum_{i=1}^{N} Y_{i,l} \oplus X_{i,l}, \tag{1}$$

where N is the number of samples, L is the number of labels, $\oplus$ represents the XOR operation and $Y_{i,l}$, $X_{i,l}$ are booleans indicating if the i^{th} prediction contains the l^{th} label.

- F1 Score: We compute the F1 score in the multi label setting as the average of the F1 scores for each of the classes. Thus the F1 Score is given by

$$F1 = \frac{1}{L}\sum_{l=1}^{L}\frac{TP_l}{TP_l + \frac{1}{2}(FP_l + FN_l)} \tag{2}$$

where L is the number of labels, TP_l, FP_l and FN_l are the number of true positives, number of false positives and number of false negatives for the l^{th} regulation respectively.

We show the results of our experiments in tables 4 and 5. As we can see the Transformer methods do a much better job than the baselines due to their ability to capture context and long dependencies well. While vanilla LSTM performs worse than some of the baselines, the hierarchical LSTM fares much better. This is primarily due to the better aggregation of individual case facts resulting in higher quality case representations. We also see as expected that the SEBI-BERT outperforms the general domain English BERT owing to its ability to capture legal domain specific language representation. Despite the scarcity and domain specificity of the data, *our best performing model, SEBI-BERT obtains an F1 score of* 0.62 *and a hamming loss of* 0.027.

Table 4: Results for traditional machine learning

Features →	Character n-grams		Word n-grams		GloVe	
Classifiers ↓	F1	HL	F1	HL	F1	HL
Linear SVM	**0.52**	0.041	0.24	0.044	0.31	0.041
RBF SVM	0.24	0.043	0.42	0.039	**0.47**	**0.033**
RF	0.46	0.040	0.19	0.045	0.30	0.040
MLP	0.41	**0.037**	**0.50**	**0.031**	0.38	0.037

5 CHALLENGES

Predicting the regulation violations is a challenging task and we summarize our key takeaways below:

- **Number of facts**: To predict whether a violation has occurred or not, we must first understand all the facts of the case.Violations are often the results of multiple facts and thus these facts must be analyzed together, which becomes harder as the number of facts increases. While approaches such as sliding windows or hierarchical methods can alleviate this problem to some extent, they suffer from insufficient long-range attentions. Facts of the case can span from a few sentences to hundreds of sentences depending on how complex the case or situation is.
- **Avalanche Effect**: Avalanche effect is the property where a small change in input results in a massive change in the output. Two case-files can have a large overlap between the facts of the case, but a single difference in the facts can result in one case-file violating and the other not violating. Consider the following example. Two companies A and B publish reports after their public issues. Due to the similar nature of events, they naturally have a similar sets of facts. But because company A published the reports within three working days and company B did it after four working days, company A was compliant whereas company B violated SEBI guidelines 7.2.1.1(a) which deals with post issue obligations.
- **Domain specificity**: Due to the domain there exists a large amount of specific terminology and jargon. In the legal domain especially, there exists strict constraints on the meanings as even a small misunderstanding can have enormous consequences. Moreover standard NLP tools that are applicable in the general domain do not perform as well, and thus domain specific tools have to be built for the tasks such as entity recognition. Entities such as "Badliwalas" i.e a financier who lends money to both buyers and sellers of shares when they are not able to pay or deliver are specific to this domain, and thus will not be recognized with off-the-shelf models.

Table 5: Results for the LSTM and Transformer based approaches

Models	Metrics	
	F1	HL
LSTM	0.26	0.054
Hierarchical LSTM	0.53	0.035
BERT	0.57	0.031
SEBI-BERT	**0.62**	**0.027**

6 CONCLUSION

In this work, we created a dataset for penalty violation by extracting regulation violation information from the SEBI case-files. We then developed a system to predict the regulations violated, with the best performing model obtaining a F1 score of 0.62. Directions of future work include experimenting with architectures that handle longer document spans better, and utilizing these signals as features for other tasks such as predicting the penalties for the detected violations.

ACKNOWLEDGMENTS

This work has been supported by J.P. Morgan AI Faculty Research Award. Any opinions, findings, and conclusions in this paper are those of the authors only and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: Pretrained Language Model for Scientific Text. In *EMNLP*. arXiv:arXiv:1903.10676

[2] Rachana Buch. 2018. A Survey on Multi Label Classification.

[3] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv* abs/1810.04805 (2019).

[5] Simon Haykin. 1994. *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

[6] Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, Vol. 1. IEEE, 278–282.

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[8] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* (Sep 2019). https://doi.org/10.1093/bioinformatics/btz682

[9] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[10] Claude Sammut and Geoffrey I. Webb (Eds.). 2010. *TF–IDF*. Springer US, Boston, MA, 986–987. https://doi.org/10.1007/978-0-387-30164-8_832

[11] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 1480–1489. https://doi.org/10.18653/v1/N16-1174