International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 05 | May 2021 www.irjet.net p-ISSN: 2395-0072

Legal Judgement Prediction System

Kavita Shirsat[1], Aditya Keni[2], Pooja Chavan[3], Manasi Gosavi[4]

[1]*Assistant professor, Dept of Computer Engineering, Vidyalankar Institute Of Technology, Maharashtra, India*
[2,3,4]*UG student, Dept of Computer Engineering, Vidyalankar Institute Of Technology, Maharashtra, India*

---***---

Abstract - *With the development and innovation of machine learning technology, more and more fields try to apply artificial intelligence to practical scenarios. We try to use a machine learning model to assist the judgment of the preliminary case results. In this paper, we analyze the basic description of the case, and apply a machine learning model to predict the possible IPC Section that will be applicable based on the fact of the case. This will also provide information such as penalty, accusation and legal provisions etc. On the one hand, the forecasting results can help the judges and lawyers to make decisions, on the other hand, it can also help the non-legal professionals to have a basic understanding and judgment of the case*

***Key Words*: Cases dataset, IPC section, Naïve Bayes, Random Forest, SVM**

1. INTRODUCTION

In the judicial system, the scope of work with text documents is very significant, and the decision-making process must always be fair and transparent. Manually processing such a volume of information is very difficult and sometimes almost impossible. In addition, people without legal education and involved in the trial are faced with many problems and issues that are difficult to solve without asking a lawyer. Courtjudgementsplayacrucialroleinlitigation,legalstudy, and court decision-making because some of them are exemplars of legal usage and interpretation. Lawyers use these judgements to analyse if their clients could win the lawsuits. In the field of Artificial Intelligence, court judgement prediction is a challenging task for following reasons. First, although legal interpretation is based on logical deduction, it is far too complex to handcraft rules and to imitate such tasks with a computational model. Also, it is difficult to obtain the public dataset of Indian cases and judgements. Even if there is some source to retrieve the onlinetext of judgements, that mainly provides for search purposes only. Therefore, some related work made their own dataset for model training and testing. Proper data analysis and model construction can avoid these problems very well. Therefore, tools to the intellectual analysis of the entire volume of information, to predict possible judicial decisions to citizens on the one hand, and to facilitate the routine workoflawyers ontheotherare required.For developing thislegaljudgementpredictionsystem wewillbe usingtextminingand machinelearningalgorithms.

2. RELATED WORK

Legal Judgment Prediction (LJP) is a promising technique that aims to provide appropriate judgment advice. LJP plays an important role in legal assistant systems, which can help legal professionals (e.g., judges, lawyers, and prosecutors) to improve their work efficiency and reduce the risk of making mistakes. Furthermore, it can benefit ordinary people who lack rich legal knowledge but desire to know the possible judgment result by describing a case they are concerned about. By exploiting the legal knowledge contained in massive law articles and case judgment documents, LJP will free people from the laborious tasks of information retrieval and data analysis.[1] This predicts the judicial decisions automatically given the fact description. The proposed method captures the dependencies by a prediction forward-propagate mechanism over a directed heterogeneous graph, and a novel prediction task, attribute prediction. The experiments prove the efficiency of the method and show the superior of our model on real-world datasets.[2] In this paper they analysed the basic description of the case and applied the deep learning model to predict the judgement results from the three aspects that are penalty, accusation and legal provisions.[3]The incorporation of attention mechanism and hierarchical sequence encoders is adopted to learn better semantic representations and interactions among different parts of case descriptions. Their approach significantly outperforms all the baseline models and achieves state-of-the-art performance on the entire LJP task.[4] The model's output shows if a person is guilty of a crime according to the facts and laws.[5] This paper implemented a legal text summarizer using a proposed model which makes use of a natural language processing technique called latent semantic analysis.[6] They presented the "Robot Lawyer" system, which aims to help participants of the legal process.[8] Evaluate the best set of features that automatically enables the identification of argumentative sentences from unstructured text.[9] In this, development process of LIRFSS is discussed. Appropriate information to be extracted for criminal cases such as date, location of occurrence and IPC (Indian Penal code) sections were determined from a sample set of documents.

Sr No.	IPC_Section	Case_No.	Petitioner	Respondent	Start_Date	End_Date	Fact	Decision
1	302, 304	195/1960	K. M Nanavati	State of Maharashtra	1959	24-11-1961	Kawas Manekshaw Nanavati, a Naval Commander, was tried for the murder of Prem Ahuja, his wife's lover by shooting him with a revolver .	Nanavati was initially declared not guilty by a jury, but the verdict was dismissed by the Bombay High Court and the case was retried as a bench trial.Later the accused was sentenced of imprisonment for life.
2	3,02,342	87/ 2007	Santosh Kumar Singh	State through CBI	1996	06-08-2010	Priyadarshini Mattoo was a 25 year old law student, who was found raped and murdered at her house in New Delhi on 23 January, 1996 by Santosh Kumar Singh who was also a student of LL.B. in campus Law Centre, Faculty of Law, University of Delhi.	In this matter the Supreme Court had commuted the death sentence awarded to prime accused Santosh Singh (son of former IPS officer), to life imprisonment .
3	302, 201	NA	Supreme Court of India	Manu Sharma	1999		A model in New Delhi working as a bartender was shot dead by the prime accused Manu Sharma, son of Congress MP Vinod Sharma by a 22 calibre pistol	Manu Sharma was sentenced to life imprisonment on 20 December 2006. On 2 June 2020 he was released from Tihar Jail by Delhi LG on grounds of good

Fig 2. Cases_1 dataset

3. METHODOLOGY

3.1 Data Collection:

For LJP, we have selected criminal category specially murder cases. The criminal case judgments were collected from the website of Supreme Court of India. The search over the website was carried out by two relevant criteria such as 'murder' keyword and time period ranging from the year 1960-2020. Numbers of search results were obtained, out of which 100 random cases were selected. The cases were collected in PDF format. Each case was about 15-20 pages long containing lots of metadata such as: petitioner name, respondent name, date of judgment, name of judges involved in the case, date, place of occurrence of crime, IPC (Indian Penal Code) Sections under which case was registered and judgment. Most of the information was useless but relevant information was used later to carry out the analysis.

3.2 Constructing Dataset:

The dataset of murder cases was built for evaluating our prediction model. With manual efforts and text summarization tools we extracted some information and created short summaries from lengthy murder cases, resulting in this dataset. It consists of two data tables called Cases and Law. The former stores all judgements and case information, and the latter stores all legal provisions regarding the former. We designed it as two tables because of two reasons. First, Judgement and Law have a many-to- many relationship. For example, if the defendant is accused of murder while attempt to kidnaping, this legal case will be judged under Indian Penal Code Section 302, and 366. Therefore, many provisions are applied to the only one judgement. Whereas Section 302 is referred in every Indian judgement which are relevant to a charge of murder. Second, it is easier to maintain each legal article's information when some of them are revised after some time.

Cases_1 Dataset:

This dataset included the information about the 100 murder cases. By creating the short summaries from lengthy pdf judgements we extracted the main attributes that would be sufficient to provide short description of any murder case. The attributes are case_no, IPC_section, petitioner_name, start _date, respondent_name, end_date, Fact, Decision. Fact of the case described the actual cause of the case. Decision of the case stated the final outcome of the case. This dataset is used to provide summary of the murder cases. Fig 2 represents the Cases_1 dataset.

Cases_2 Dataset:

Since it is a complete textual dataset with lots of vagueness and also unnecessary data, data cleaning was necessary. We manually annotated each record/case by the following procedure. First, we read each judgement's short summarized verdict and separated all issues based on a number of crimes. Hence, if the verdict had multiple issues, each of them would become a distinct record in this table. We tried to limit a case with maximum two major crimes. Then, we tried to label these issues, so in all eleven major issues were labelled. Also, we found that from the IPC sections such as 302, 304, 307, 308 any of them will be mandatorily charged for the case which are relevant to a charge of murder. Hence one attribute named was added which contained the labels based on the grouping performed on above mentioned IPC sections depending upon the fact of the case. Next, we defined as an attribute with male-0 and female-1, since many of the IPC sections are associated with crime against women. Finally, we assigned twenty-three target labels by performing grouping of the cases and following the law structure. Figure 2 represents the Cases_2 dataset.

	A	B	C	D	E	F
1	case no	Attribute	Women?	Reason1	Reason2	Lable
2	1	6	0	-1	-1	1
3	2	0	1	2	-1	9
4	3	0	0	-1	-1	0
5	4	7	0	-1	-1	2
6	5	0	0	-1	-1	0
7	6	0	0	-1	-1	0
8	7	0	0	-1	-1	0
9	8	5	0	5	8	3
10	9	0	0	-1	-1	0
11	10	5	0	5	10	4
12	11	2	0	-1	-1	5
13	12	0	1	4	-1	0
14	13	0	0	-1	-1	0
15	14	0	1	-1	-1	0

Fig 1. Cases_2 dataset

Law Dataset:

This table provides all detail of every IPC Section mentioned in Cases_1 dataset. attribute indicating the section no and attribute containing the explanation of the corresponding IPC section. Over all this dataset contained 50 records (IPC sections related to murder and related issues).

	IPC_Section	Title	Description.
2	27	Property in possession of wife, clerk or servant	A person employed temporarily or on a particular occasion in the capacit
3	32	Words referring to acts include illegal omissions	In every part of this Code, except where a contrary intention appears from the c
4	34	Acts done by several persons in furtherance of common intention -	When a criminal act is done by several persons in furtherance of the common in

3.3 Model Processing:

A. Model's Input: Our model's input is the text of facts from a legal murdercase.

B. Model'sTraining:Fortrainingourmodelwetriedsome traditional machine learning algorithms.

1. Naïve BayesClassifier:

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. In statistics, naïve bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' Theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian Network models, but coupled with kernel density estimation, they can achieve higher accuracy levels.

Abstractly,naïveBayesisaconditionalprobabilitymodel given as,

$$p(C_k \mid x_1, \dots, x_n) = \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

where the evidence $Z = p(\mathbf{x}) = \sum_k p(C_k)\, p(\mathbf{x} \mid C_k)$ is a scaling factor dependent only on $x_1, \dots, x_n$, that is, a constant if the values of the feature variables are known.

The naïve Bayes classifier combines this model with adecision rule. One common rule is to pick the hypothesis that is most probable; this is known as the *maximum a posterior* or *MAP* decisionrule.Thecorrespondingclassifier, aBayes classifier, is the function that assigns a class label $\hat{y} = C_k$ for some k as follows:

$$\hat{y} = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}}\ p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).$$

We have used Gaussian Naive Bayes for training our model. Gaussian naïveBayes:

When dealing with continuous data, a typical assumptionisthatthecontinuousvaluesassociatedwith each class are distributed according to a normal (or Gaussian) distribution. For example, suppose the training data contains a continuous attribute, *x*. We first segment the data by the class, and then compute the mean and variance of *x* in each class. Let μ_k be the mean of the values in x associated with class C_k, and let σ_k^2 be the Bessel Corrected Variance of the values in *x* associated with class C_k. Suppose we have collected some observation value *v*. Then, the probability *density* of *v* given a class C_k, $p(x = v \mid C_k)$, can be computed by plugging *v* into the equation for anormal distribution parameterized by μ_k and σ_k^2. Thatis,

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}}\, e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

Using Gaussian naive Bayes classifier we have achieved an accuracy of almost 74%. The below fig represents the classification report of the gaussian naive Bayes classifier.

	precision	recall	f1-score	support
0	1.00	0.86	0.92	7
4	0.50	0.50	0.50	2
5	1.00	1.00	1.00	1
7	0.50	1.00	0.67	1
9	1.00	1.00	1.00	1
10	0.00	0.00	0.00	1
13	0.50	1.00	0.67	1
14	1.00	1.00	1.00	1
17	0.00	0.00	0.00	1
20	0.00	0.00	0.00	1
21	0.00	0.00	0.00	0
22	0.67	1.00	0.80	2
micro avg	0.74	0.74	0.74	19
macro avg	0.51	0.61	0.55	19
weighted avg	0.70	0.74	0.70	19

Fig 3.

2. Random Forest:

A random forest algorithm is a collection of tree-structured classifiers. The random forest algorithm trains a number of trees with slight variation in the subsets of data. A case is added to each subset containing random selections from the range of each variable. The random forest algorithm has relatively high accuracy among algorithms for classification. It can handle large data sets and it has features to balance the unbalanced data. Using this classifier we achieved accuracy of around 68%. Below figure represents the classification report of random forest.

	precision	recall	f1-score	support
0	0.89	0.89	0.89	9
7	0.50	0.33	0.40	3
8	1.00	1.00	1.00	2
9	0.00	0.00	0.00	0
10	0.00	0.00	0.00	1
15	1.00	1.00	1.00	1
19	1.00	0.50	0.67	2
21	0.00	0.00	0.00	1
22	0.00	0.00	0.00	0
micro avg	0.68	0.68	0.68	19
macro avg	0.49	0.41	0.44	19
weighted avg	0.76	0.68	0.71	19

Fig 4.

With few features and few trees, the algorithm scores poorly, the f1 value increases for some number of trees and decreases for some number of trees for the same number of features.

3. Support Vector Machine (SVM):
Support Vector Machines are linear classifiers that construct a hyperplane with the largest margin between the positive and negative examples to reduce the error of the classifier. Let us suppose that, the set S has n training examples:

$$S = (x_1, y_1), \ldots\ldots (x_n, y_n)$$

where $x_i \in \mathbb{R}^n$ and $y_i \in \{+1, -1\}$, indicating that xi is a positive or a negative example respectively. Then, the equation of the separating hyperplane can be represented by

$$< w.x_i > +b = 0 \quad where \quad w \in \mathbb{R}^n, b \in \mathbb{R}$$

We achieved an accuracy of about 63% by using Grid Search. Below figure represents the classification report of SVM.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	0.00	0.00	0.00	1
2	0.00	0.00	0.00	1
4	0.00	0.00	0.00	1
5	0.00	0.00	0.00	0
7	0.33	1.00	0.50	1
8	0.33	0.50	0.40	2
18	0.00	0.00	0.00	1
21	0.00	0.00	0.00	1
23	0.00	0.00	0.00	1
micro avg	0.63	0.63	0.63	19
macro avg	0.17	0.25	0.19	19
weighted avg	0.58	0.63	0.59	19

Fig 5.

With fewer features and lower values of C, the f1 value is also low, but as the value of C is increased, the performance increases. However, we found that this is true only up to C = 0.1; after that, as we increase the value of C, the f1 value slightly decreases. When applying statistical tests, we found that there is no significant difference between the results obtained with C∈{0.1,1}, therefore we considered taking C = 1.

On comparing the results of naive Bayes, Random Forest, and SVM, we found the results of naive Bayes to be quite satisfying with the highest accuracy of 74%.

Hence naive Bayes classifier model was finalized for prediction.

3.4 User Interface:

We had developed the website as our user interface with user login and authentication by OTP verification. Along with the prediction of the IPC section, we provide the feature of viewing the case summary, IPC section information, and also analysis of cases. For developing the website we have used NodeJs, in which we have created different routes for creating different pages with the help of the render method. Also, we have made use of inbuilt npm packages in NodeJS like express, ejs, body-parser, passport, mongoose which helped us to simplify the process. The passport package really helped us to simplify the login and sign-up process for user authentication. For the design part of our website, we have took helped of Bootstrap 4 components like Navbar, Jumbotron, Jumbotron-fluid, Cards, Buttons, Grid Functionality which helped us to make our website more interactive. Moreover, we have made use of the Font Awesome library which helped us to add user-friendly and interactive icons.

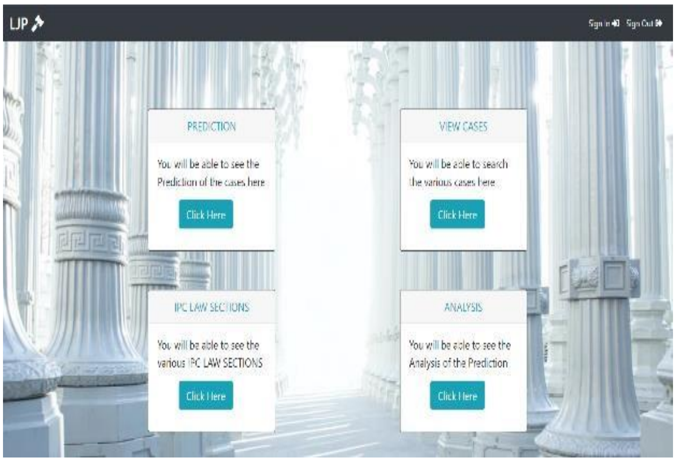

Fig 6.

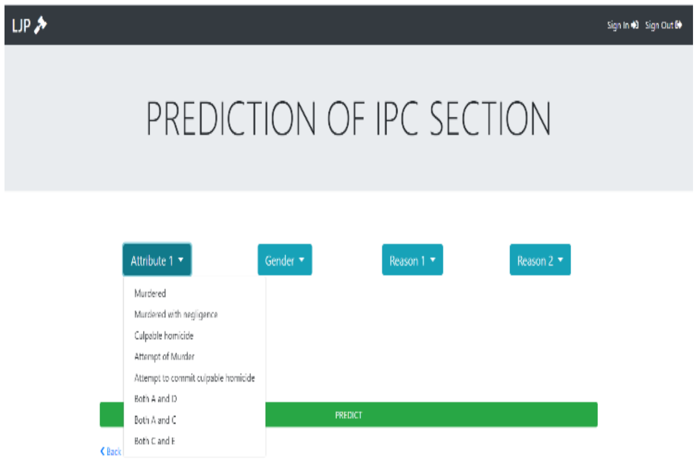

Fig 7.

4. CONCLUSIONS

In this paper, we offer a prediction model for Indian legal cases particularly murder cases. We have trained our model on various traditional machine learning techniques like Naive Bayes, Random Forest and SVM. In comparison we found that Naive Bayes provided satisfactory results with the highest accuracy of 74% as compared to the other two techniques. Also, we have prepared a Cases and Law datasets which consist of 100 murder cases and 50 law sections respectively from lengthy cases which were in the pdf form. In the future, we will focus on improving the performance of our model with other variations of Machine learning. Also, we will consider utilizing more records in our cases dataset and build the model that can practically represent the sophisticated criminal law structure.

REFERENCES

[1] L. Chen, N. Xu and Y. Wang, "Legal Judgment Prediction with Label Dependencies," 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calgary, AB, Canada, 2020, pp. 361-365

[2] B. Chen, Y. Li, S. Zhang, H. Lian and T. He, "A Deep Learning Method for Judicial Decision Support," 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Sofia, Bulgaria, 2019, pp. 145- 149.

[3] S. Li, H. Zhang, L. Ye, X. Guo and B. Fang, "MANN: A Multichannel Attentive Neural Network for Legal Judgment Prediction," in IEEE Access, vol. 7, pp. 151144-151155, 2019.

[4] K. Kowsrihawat, P. Vateekul and P. Boonkwan, "Predicting Judicial Decisions of Criminal Cases from Thai Supreme Court Using Bi-directional GRU with Attention Mechanism," 2018 5th Asian Conference on Defense Technology (ACDT), Hanoi, 2018, pp.50-55.

[5] K. Merchant and Y. Pande, "NLP Based Latent Semantic Analysis for Legal Text Summarization," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, 2018, pp.1803-1807.

[6] Khasianov, I. Alimova, A. Marchenko, G. Nurhambetova, E. Tutubalina and D. Zuev, "Lawyer's Intellectual Tool for Analysis of Legal Documents in Russian," 2018 International Conference on Artificial Intelligence Applications and Innovations (IC-AIAI), Nicosia, Cyprus, 2018, pp. 42-46.

[7] L. T. Barco Ranera, G. A. Solano and N. Oco, "Retrieval of Semantically Similar Philippine Supreme Court Case Decisions using Doc2Vec," 2019 International Symposium on Multimedia and Communication Technology (ISMAC), Quezon City, Philippines, 2019, pp. 1-6.

[8] P. Poudyal, T. Goncalves and P. Quaresma, "Experiments on identification of argumentative sentences," 2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA), Chengdu, 2016, pp. 398-403

[9] D. Gaur, "Data mining and visualization on legal documents," 2011 International Conference on Recent Trends in Information Systems, Kolkata, 2011, pp. 132-136.