IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

IDENTIFICATION OF IPC FOR POLICE COMPLAINT USING NLP

[1]Mr.Bhushan Nandwalkar, [2]Kirtish Wankhedkar, [3]Neha Yeolekar, [4]Upasana Patil

[1]Computer Engineering,

[1]Shri Vile Parle Kelavani Mandal's Institute of Technology, Dhule, India.

Abstract: Identification of IPC (Indian Penal Code) system is the system where the system can identify the IPC's on the basis of structured or un-structured data without human interference. There are many disadvantages which may arise due to unawareness about the law. As, the India has largest democracy where polices places important roles in providing justice and punishment, hence it is essential for them to know the IPC sections which must be imposed on specific charge. Our Indian Constitutional law is huge and complex, everyone cannot spend their time in reading and understanding it, even law professionals find it complex to understand it. Therefore, the purpose of our research paper is to present or propose such a system which will be helpful for the polices to impose the IPC sections on criminals or rule breakers without interference of lawyers as this process can be lengthy and time consuming. Here, the system uses Universal Sentence Encoder as a model. It predicts the different IPC sections on the basis of the given input. In this way, it helps to identify the appropriate amendment law (IPC). In this, by using Natural Language Processing (NLP) and Deep learning(DL) produce accurate results.

***IndexTerms*:** Indian Penal Code (IPC), Natural Language Processing (NLP), Deep Learning (DL), Transfer-Learning, Universal Encoder Model.

I. Introduction

The Indian Penal Code (IPC) is the document which covers all the criminal activities along with their punishments which must be charged with. The objective of enabling IPC was to provide discrimination free justice. It is applicable to each and every person of India. This systems helps the polices to charge appropriate IPC on every individual breaking the rules.

It has all the aspects of criminal laws apart from these also, one can get law codes. But, these cannot be considered viable. Therefore, it is required to know about laws and act accordingly. Informing about law most correctly is the important role of any system related to law.

Earlier, there were many systems proposed but they have some limitations. [5] Due to this, we have suggested this method which is most effective than all of them; here we tried to remove all the limitations which were present in previous systems using N.

II. RELATED WORK

[1].In this paper, The author use bags of word technique to extract keyword and label with belonging IPC section and use Convolution Neural Network to train the classification model. And in output layer use Soft Max function to predict the IPC sections.

[2].The documentation is organized in tree data structure where similar keyword are map together and having ID for particular mapping keyword. In learning process machine get reward for total matching keyword from quires and from documentation. And user will rate the answer in which machine will improve itself.
[3].The author retrieve the keyword from query and match with the keyword belongs to the documents and extract the specific answer from retrieve documents through the question classification helps.

[4].As part of the abstract, each sentence is labeled with its role using one of the following classes: background, objective, method, result, or conclusion. Existing models for sentence classification based on artificial neural networks (ANNs).

[6].In these, words with similar meaning are found. Grouping of these words takes place and the query is retrieved using indexing; ranking is done on the basis of learning.

III. CHALLENGES IN PREVIOUS SYSTEMS

a) Semantic nature type question was confusing to system: The previous model have problems to identify the similar words or the words which have similar meaning for example to make prediction on the murder cases, the user shall use 'kill' word instead of murder and if machine or model cannot detect those similar words, it will predict wrong IPC on it. To address this issue, we incorporate a word embedding strategy into the model, which aids in the identification of comparable words with similar meanings. Word embeddings often represent words as a real-valued vector that encodes the meaning of the word, with words that are near in the vector space anticipated to have comparable meanings. The learning process is either supervised, using document statistics, or unsupervised, employing a neural network model for some purpose, such as document categorization.

b) The accuracy of predicting a verdict falls as the number of charges increases- Human as well as machine can predict wrong answer but humans has ability to relearn its mistake and update knowledge, so we try to implement this concept in model also. Whenever our model predict wrong IPC, the right IPC with there respective case will save in training data where our model will try to update its weights. So, next time on same cases our model will predict correct IPC. This is done using the transfer learning approach.

IV. PROPOSE METHOD

1. Universal sentence encoder

Text Embedding is crucial in the development of any Deep Learning and NLP model. Text Embedding, commonly referred to as Word Embedding, is a method of converting words or phrases into numerical vectors. A numerical vector is an array of the number of a specific 4x1 size vector consisting of four numbers. Assume that each of the four numbers has a point in 4D space, and that there are two vectors in 4D space. As a result, we can determine how near or far apart the two vectors are based on the distance between them.

As a result, turning data into a vector is important since machine learning research requires a lot of effort after data is turned into a vector Along with their words, the context of the entire phrase must be recorded in that vector when embedding a sentence. The "Universal Sentence Encoder" comes in helpful in this situation.

Text is encoded into high-dimensional vectors using the Universal Sentence Encoder, which may be used for text categorization, similarity measures, grouping, and other speech recognition applications. TensorFlow-hub provides a pre-trained Universal Sentence Encoder openly accessible. It comes in two flavors: one with a Transformer encoder and another with a Deep Averaging Network (DAN). There is a trade-off between accuracy and computing resource needs in both forms. The one with the Transformer encoder is computationally more expensive, despite having improved precision. The DNA encoding approach is less expensive to compute and has a lesser precision.

Let's now examine each component of Universal Sentence Encoder in more detail.

(1) Tokenization: Tokenization begins with converting the sentences to lowercase and tokenizing them using the Penn Treebank tokenizer.

(2) Encoder: This is the component that encodes a sentence into 512-dimension embeddings of fixed length.

A) TRANSFORMER ENCODER

The encoder part of the original transformer architecture is used in this variant. This variant has six layers of transformers. An attention module is followed by a feed-forward network in each layer.

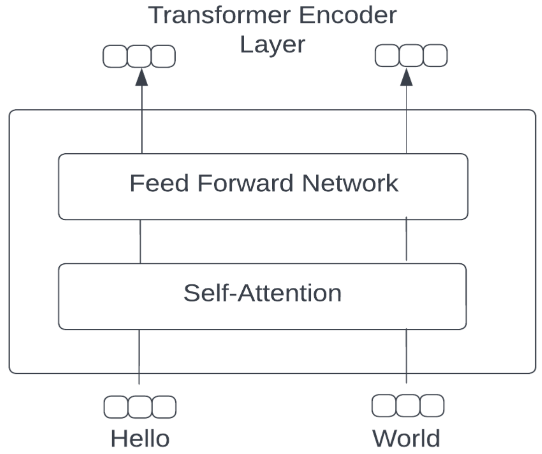

Figure 1 Transformer layer Architecture

Self-attention generates word representations based on word order and surrounding context. The output context-aware word embeddings are added element by element and then divided by the square root of the sentence length in order to account for the variation in sentence length. As an output sentence embedding, we receive a 512-dimensional vector.

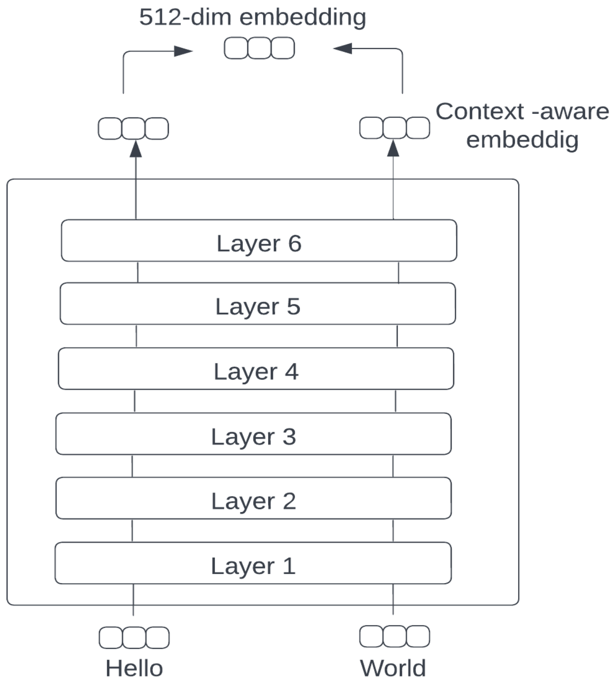

Figure 2 Transformer layer Architecture

Because of its complicated design, this encoder provides better accuracy on downstream jobs but uses more memory and computation resources. Also, because self-attention has O(n2) time complexity, the compute time scales drastically with the length of the text. However, it is just slightly slower for short phrases.

B) DEEP AVERAGING NETWORK (DAN)

To begin, all of the words and bi-grams in a phrase's embeddings are averaged. After that, they're put into a four - layer feed-forward deep DNN to generate a 512-dimensional phrase embedding. The embeddings for words and bi-grams are learnt during training.

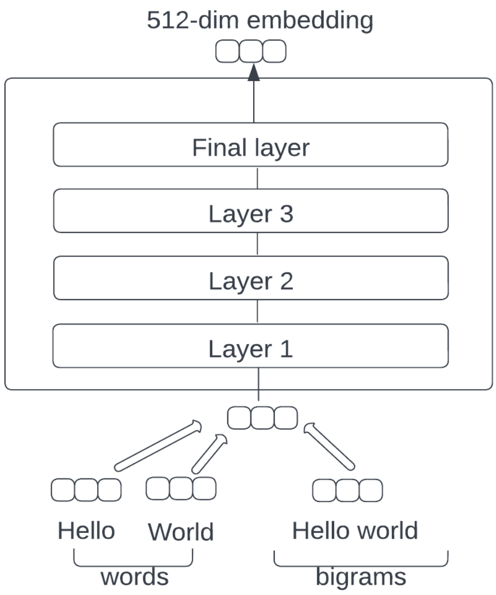

Figure 3 DAN Architecture.

It has somewhat lower accuracy than the transformer type, but it has a much faster inference time. The compute time is linear in terms of the length of the input sequence because we are simply executing feedforward operations.

2. SYSTEM ARCHITECTURE

This system comprises four modules-

a) Module A

b) Module B

c) Module C

d) Module D

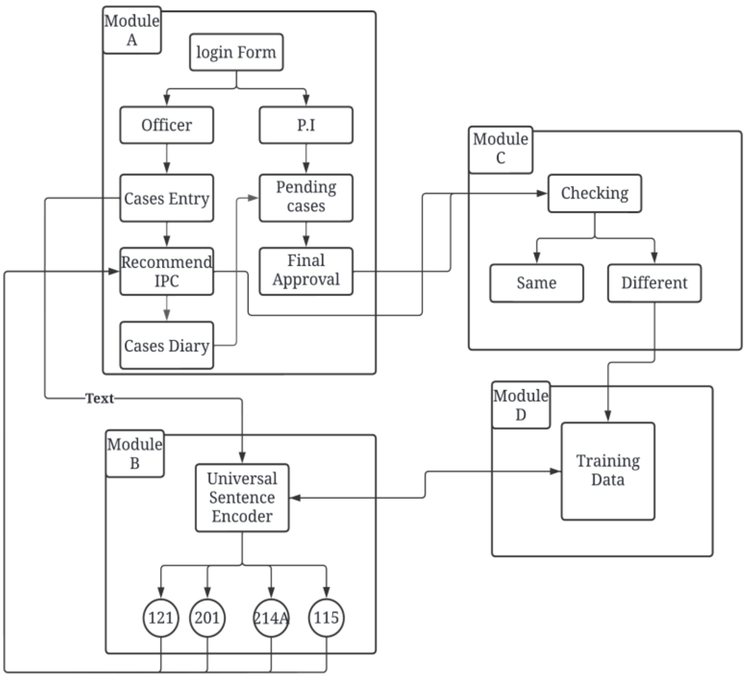

Figure 4 System Architecture.

- Module A and Module B

Consists of login information. There are two users: Officer, and PI(Police Inspector). When the Officer login uses proper credentials and enters the cases into the system. This data is then given to Module B which contains Universal Sentence Encoder.

This encoder converts the text into high dimensional vectors and predicts IPC on given cases and the predicted IPC is passed in recommended IPC slot. Where officers can also edit them by adding additional IPC or deleting some IPC which are not approved on cases. After that cases diary is formed in which whenever P.I login he can see those pending cases diary whether to approve the cases or not. P.I also can make changes in IPC based on his/her knowledge.

- Module C

This module helped system to relearn from its mistake.

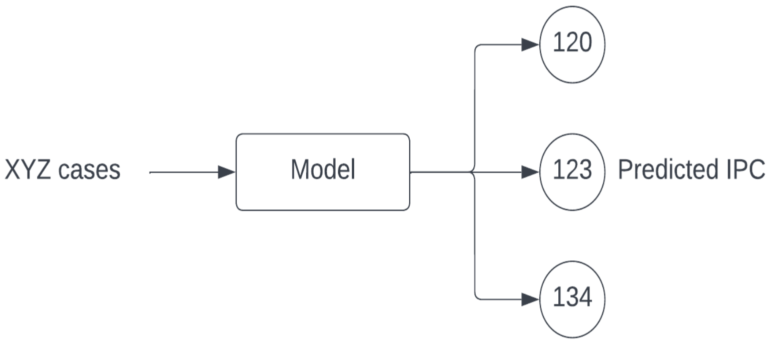

Figure 5 Predicted IPC by Model.

For example, let's say that model predictsed IPC on XYZ cases are 120,123,134 but officers or P.I based on their knowledge think in XYZ cases instead of 134 it should be 302 so they make a change in recommended slot. Here we can see.

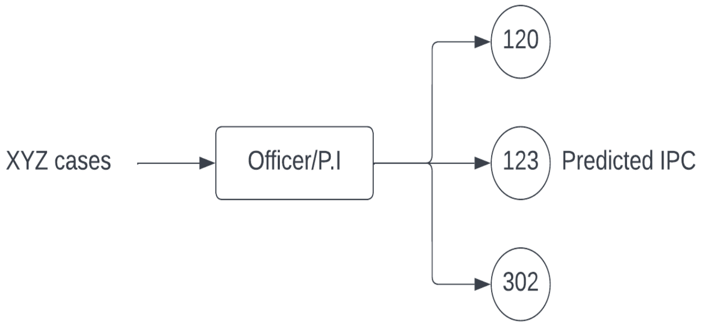

Figure 6 Predicted IPC by Officer/P.I.

Figure 7 Predicted IPC equation

Prediction done by the model are not similar to the prediction by Officer or P.I in which Model C will then pass this Officer/P.I preds IPC with their respective case to Training data.

- *Module D*

Through the training data the model will try to update its weight so the next time similar cases will come then it will predict the right one.

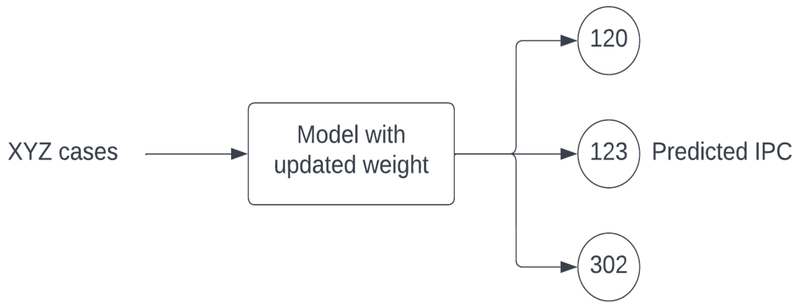

Figure 8 Updated Model Predicted IPC

If both predicted and confirmed are similar, then training of data will not take place. All the process is repeated, to get the required output.

V. Experiments

1. Data Collection

The data obtained is based on information provided publicly available by the Indian Courts. In this experiment, every infraction under the IPC is recorded, together with the relevant section and penalty. The Indian courts publish these IPC records, that are available on any Indian court's website.

B	C	D
Section	Offence	Punishment
["109 IPC"]	Abetment of any offence, if the act abetted is committe	Same as for offence abetted
["110 IPC"]	Abetment of any offence, if the person abetted does the	Same as for offence abetted
["111 IPC"]	Abetment of any offence, when one act is abetted and a	Same as for offence intended to be abetted.
["113 IPC"]	Abetment of any offence, when an effect is caused by tl	Same as for offence committed
["114 IPC"]	Abetment of any offence, if abettor is present when offe	Same as for offence committed
["115 IPC"]	Abetment of an offence, punishable with death or impri	Imprisonment for 7 years and fine.
["116 IPC"]	Abetment of an offence, punishable with imprisonment	Imprisonment extending to a quarter part of t
["117 IPC"]	Abetting the commission of an offence by the public or	Imprisonment for 7 years, or fine, or both.
["118 IPC"]	Concealing a design to commit an offence punishable w	Imprisonment for 3 years, or fine, or both.
["119 IPC"]	A public servant concealing a design to commit an offer	Imprisonment extending to half of the longest
["120 IPC"]	Concealing a design to commit an offence punishable w	Imprisonment extending to a quarter part of t
["121 IPC"]	Waging or attempting to wage war or abetting the wag	Death, or imprisonment for life and fine.
["121A IPC"]	Conspiring to commit certain offences against the State	Imprisonment for life, or imprisonment for 10
["122 IPC"]	Collecting arms, etc., with the intention of waging war a	Imprisonment for life, or imprisonment for 10
["123 IPC"]	Concealing with intent to facilitate a design to wage wa	Imprisonment for 10 years and fine.
["124 IPC"]	Assaulting President, Governor, etc., with intent to com	Imprisonment for 7 years and fine.

Figure 9 Screen shot of Training data

In the screenshot of the database, the Section column contains all IPCs, and the Offence column contains their respective offences. In addition, we have punishment columns to help the officer or P.I. see their respective punishment for a particular offense.

2. Training

To train the model, Adam gradient descent is used, updating all parameters, such as token embeddings and character embeddings. Multilabel binarization is used on train dataset for multiple prediction on IPC. One dense layer is used with a Relu activation function. And in compile process for loss use categorical crossentropy and metrics has accuracy.

3. Experiment Result

Before the model could be trained, the dataset must be created. Datasets are collections of data. A dataset can be organised, has a unstructured, structured or semi-structured, according on its nature. The whole dataset was generated from scratch because there is no pre-structured IPC-based judicial decision dataset in India or on any website. In this study, an IPC structured data collection is organised in the form of an MS-Excel file, which can retrieve data efficiently. Documents related to IPC cases in hard copy form. Because there were no soft copies of the case files were available on any website and to verify and validate the data, these data were manually sorted into MS-Excel file format due to the vast number of hard copies of the cases were present enter manually. These data will be required for the training of a deep learning model that includes NLP.

4.Transformer Encoder

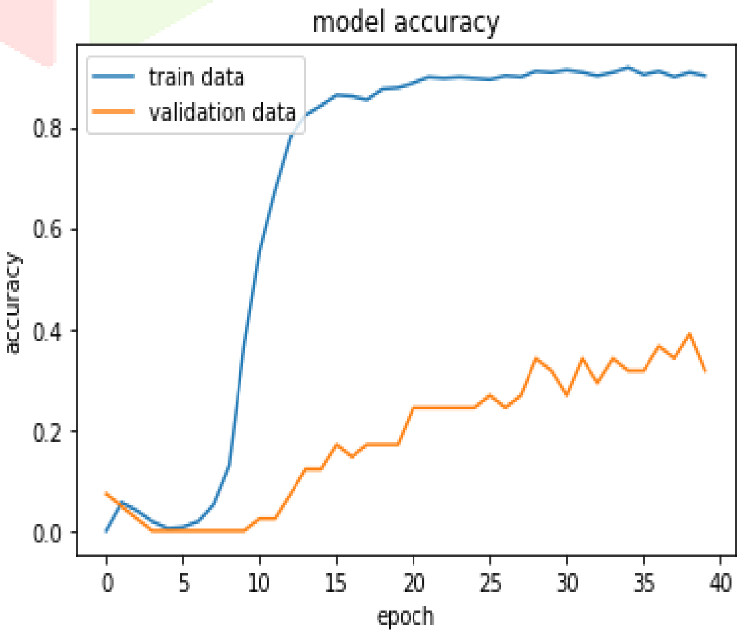

Figure 10 Transformer Encoder Accuracy graph.

Table 1 Transformer Encoder accuracy table.

Accuracy	F1-Score	Recall	Precision
73.17073	0.845070	0.7317073	1.0

5.Deep Averaging Network (DAN)

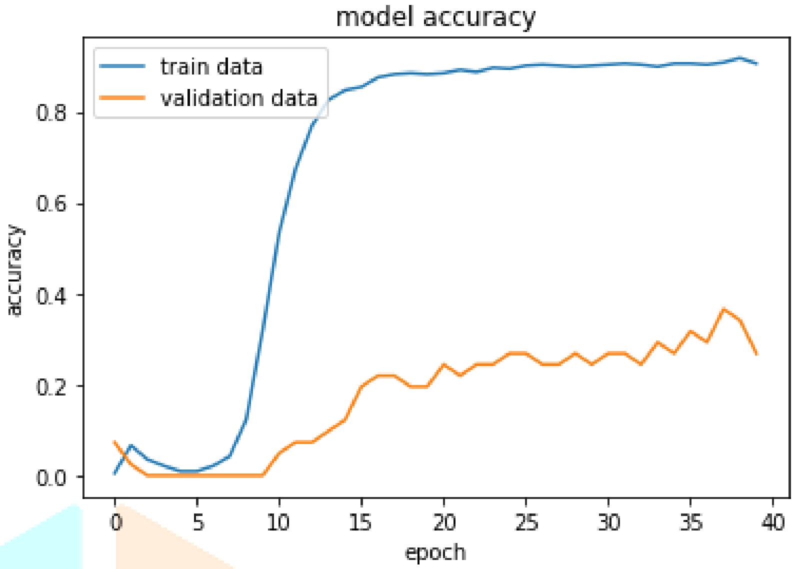

Figure 11 Deep Averaging Network Accuracy graph.

Table 1 Transformer Encoder accuracy table.

Accuracy	F1-Score	Recall	Precision
80.48780	0.89189	0.8048780	1.0

In this comparison of variants, we see that variant 2: DAN architecture performs better than variant 1: Transformer Encoder. From the following graph, we can see that 25 epochs of training is sufficient for both models.

VI. Conclusion

In the traditional system, lawyers play a crucial role in recognizing IPC. The introduction of NLP and DL to this system makes things simpler for cops to assign several IPC sections to criminals and rule violators. Furthermore, this technique is accurate, and unbiased decisions are made in a short period of time. This system does not require any human-engineered characteristics, and that might be used in both structured and non-structured cases by user. It will contribute to a society that is free of corruption and discrimination. This system is unbiased because it can provide approximate accurate decision.

VII. Acknowledgment

The authors are thankful to Prof. Bhushan.R.Nandwalkar, for their encouragement, support and guidance for successful completion of the paper.

References

[1] V Gokul Pillai, Lekshmi R Chandran, “Verdict Prediction for Indian Courts Using Bag of Words and Convolutional Neural Network”, International Conference on Smart Systems and Inventive Technology (ICSSIT 2020), IEEE proceddings, 2020.

[2] D Sangeetha , R Kavyashri , S Swetha and S Vignesh, “Information Retrieval System for Laws”, IEEE Eighth International Conference on Advanced Computing (ICoAC), 2016.

[3] Rohini P. Kamdi, Avinash J. Agrawal, “ Keywords based Closed Domain Question Answering System for Indian Penal Code Sections and Indian Amendment Laws”, I.J. Intelligent Systems and Applications, 2015, PP. 12, 57-67

[4] Franck Dernoncourt, Ji Young Lee, Peter Szolovits, “Neural Network for Joint Sentence Classification in Medical Paper V Abstracts”, Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 694–700**.**, April 3-7, 2017.

[5] The Indian Penal Code is available in https://india.gov.in/indian-penal-code-act-1860

[6] WordNet: An electronic lexical database, MIT Press is available in https://mitpress.mit.edu/books/wordnet