Automatic Charge Identification from Facts: A Few Sentence-Level Charge Annotations is All You Need

Shounak Paul
IIT Kharagpur
shounakpaul95@kgpian.iitkgp.ac.in

Pawan Goyal
IIT Kharagpur

Saptarshi Ghosh
IIT Kharagpur

Abstract

Automatic Charge Identification (ACI) is the task of identifying the relevant legal charges given the facts of a situation and the statutory laws that define these charges, and is a crucial aspect of the judicial process. Prior works focus on learning charge-side representations by modeling relationships between the charges, but not much effort has been made in improving fact-side representations. We observe that only a small fraction of sentences in the facts actually indicates the charges. We show that by using a very small subset ($< 3\%$) of fact descriptions annotated with sentence-level charges, we can achieve an improvement across a range of different ACI models, as compared to modeling just the main document-level task on a much larger dataset. Additionally, we propose a novel model that utilizes sentence-level charge labels as an auxiliary task, coupled with the main task of document-level charge identification in a multi-task learning framework. The proposed model comprehensively outperforms a large number of recent baseline models for ACI. The improvement in performance is particularly noticeable for the rare charges which are known to be especially challenging to identify.

1 Introduction

In countries following a Civil Law system, there exist written statutory laws defining 'charges' (e.g., *theft*, *murder*, *sexual offence*), which one must consider before establishing a particular charge in a given situation. Charge identification is a tedious and time-consuming task requiring legal expertise, since the written laws are often described abstractly to encompass the wide-ranging scenarios in which a charge can apply. Thus automating the charge identification task can benefit the police and law practitioners.

The task of Automatic Charge Identification (ACI, also referred to as 'charge prediction') aims to determine the possible charges that might have occurred in a given situation, using automated methods (Hu et al., 2018; Wang et al., 2018; Wang et al., 2019). A method for ACI takes as input (i) a document containing a textual description of a situation, called 'fact-document' or 'facts', and, (ii) textual descriptions of the charges/charges, available as written laws (statutes) in the legal statues of a country. The method outputs the set of charges/charges committed in the given situation.

Most efforts in this task have been devoted towards distinguishing between rare and confusing charges, or learning better representations of the charge descriptions. For instance, Hu et al. (2018) designed 10 discriminative attributes to disambiguate confusing charge pairs, while Wang et al. (2018) and Wang et al. (2019) exploit pairwise co-occurrence and hierarchies of legal statutes respectively, to learn better representations for the statute descriptions. However, *there has not been much effort in learning better fact-side representations*. Fact descriptions are usually long and contain a lot of background information – e.g., the relationship between the parties involved, past disputes, etc. – which do *not* play a significant role in establishing the charges. Importantly, a lot of sentences in the fact descriptions are 'noisy' from the perspective of ACI, i.e., do not indicate any charge at all (see the example in Table 4 later in the paper). None of the prior works for ACI have considered the impact of noisy sentences or tried to identify indicative words/sentences in the facts.

Proceedings of the 28th International Conference on Computational Linguistics, pages 1011–1022
Barcelona, Spain (Online), December 8-13, 2020

In the present work, we hypothesize that *identifying charges at the sentence-level* as an auxiliary task can help to better distinguish indicative sentences from noisy ones, and learn better fact-side representations for the main task of document-level charge identification. To this end, we choose a small fraction of the fact-documents and annotate each sentence with the charges the particular sentence possibly indicates, in consultation with legal experts. We show that such sentence-level annotations, even for a small fraction of fact-documents, can help various ACI models to better identify charges for the facts.

Further, we propose a novel attention-based model with multi-task learning framework which, given the facts of a situation/case, tries to predict the relevant charges for each sentence and the overall situation (i.e., for the entire facts). Our model constructs intermediate fact-side sentence representations which are used to match with the charge-side representations to predict sentence-level charges. We also propose a novel scheme of assigning weights to each sentence based on the sentence-level predictions. Using these weights, we aggregate the sentence representations to obtain the fact-side document representation, used for document-level charge classification.

We conduct experiments on the facts stated in Indian Supreme Court judgment documents to identify charges out of a set of 20 most frequently occurring charges in the Indian Penal Code. We compare the proposed model with several state-of-the-art ACI baselines (Luo et al., 2017; Hu et al., 2018; Wang et al., 2018; Wang et al., 2019). Our proposed model provides statistically significant improvements over all the baselines, especially for the rare charges.

In brief, our contributions are as follows: (1) Different from prior works on ACI, we focus on learning better fact-side representations by annotating a small set of documents with sentence-level charges. (2) We show that, by utilizing both sentence and document-level information under multi-task learning on a small annotated subset ($< 3\%$ training examples), performance improves across a large number of ACI models, as compared to when training these models on a much larger dataset containing only document-level charges. The greatest increment in performance is on the rare classes (charges) that are known to be especially challenging to identify. (3) We propose a novel model that utilizes multi-task learning and a unique weighting scheme to learn better fact-side representations. We show that our proposed model outperforms a large number of baselines, with an improvement of 8.6% in terms of macro-F1 over the closest baseline (Wang et al., 2019). The implementation of our proposed model and the dataset used in this work are available at `https://github.com/Law-AI/automatic-charge-identification`.

2 Related Work

Early approaches: Early attempts for identifying charges or supporting laws involved analyzing legal cases in specific scenarios using mathematical and statistical models (Kort, 1957; Ulmer, 1963; Segal, 1984), where hand-crafted rules were used to make the predicted results interpretable. With the advent of Machine Learning, researchers began to model ACI as a text classification task. For instance, Liu and Hsieh (2006) try to identify cue phrases to improve classification efficiency, while Aletras et al. (2016) use n-gram bag-of-words vectors with Linear Support Vector Classifier. Lin et al. (2012) use machine learning approaches to label documents, classify the category of the case and predict the judgment.

Attentive Deep Learning Models: The early methods use manually designed, shallow textual features, which require substantial human effort, as well as domain knowledge. The success of neural models in various legal NLP tasks (Zhong et al., 2020) has motivated the use of attention-based deep learning models for ACI as well. For instance, Luo et al. (2017) dynamically generate representations for charge texts attentively, based on the fact representations. Hu et al. (2018) design 10 discriminative legal attributes, such as whether the charge is intentional, whether there is some physical injury, etc. These facts are encoded and attentively used to predict these attributes and charges as multi-task learning. For identifying the relevant law articles, Wang et al. (2018) propose DPAM, which exploits the fact that some articles co-occur frequently due to high semantic similarity in their textual descriptions. Whereas, Wang et al. (2019) use the tree-like hierarchies that exist among law articles to first identify the parent-laws and then the children-laws. Better representations are learnt for rare children-laws by considering the semantics of the siblings. In one of the rare works on English, Chalkidis et al. (2019) try out different methods, such as Bi-GRU with Attention, Hierarchical Attention Network and BERT for identifying charges in judgment cases from the European Court of Human Rights (ECHR). We consider all the above mentioned neural models as baselines in our work.

Modeling multiple legal tasks: Recently, a few large legal datasets such as the CAIL Judgment Prediction dataset (Xiao et al., 2018) have become available which contains judgments from the Supreme People's Court of China, and annotations for three sub-tasks – predicting legal articles, charges and prison terms. Some works have utilized correlations between these sub-tasks. For instance, Zhong et al. (2018) note that an acyclic dependency exists between the sub-tasks, while Yang et al. (2019) introduce a multi-perspective forward prediction and backward verification framework to utilize result dependencies between the sub-tasks. Xu et al. (2020) use graph-based methods to group statutory articles into communities, and use distinguishable features from each community to attentively encode facts. Since these methods are geared toward utilizing the correlations between the related sub-tasks (and need training data pertaining to all these tasks), we do not consider them as baselines, since our main focus is only charge identification.

3 Proposed model for Automatic Charge Identification

Problem Definition: The ACI task takes as input (i) a textual description of the facts (of a situation), and (ii) textual descriptions of a set of charges. The fact description can be considered as a sequence of sentences $x = [x_1, x_2, \ldots, x_n]$, where each sentence is a sequence of words $x_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$. The set of charges is $Y = [y_1, y_2, \ldots, y_C]$, and each charge has a corresponding charge description text, which can also be seen as a hierarchical sequence of sentences and words. The main task of document-level ACI is to identify one or more charges from Y, given x and Y. The auxiliary task of sentence-level ACI (that we consider in this work) is to identify zero or more charges given each x_i and Y.

Proposed model: Figure 1 gives an overview of our proposed model. We first obtain a fact-side representation (which can be either a representation of the whole fact, or of each sentence in the fact). Then we use this fact-side representation to dynamically generate the charge-side representation, and then pass these through a fully connected layer to obtain the final predictions. Below, we describe each of these components.

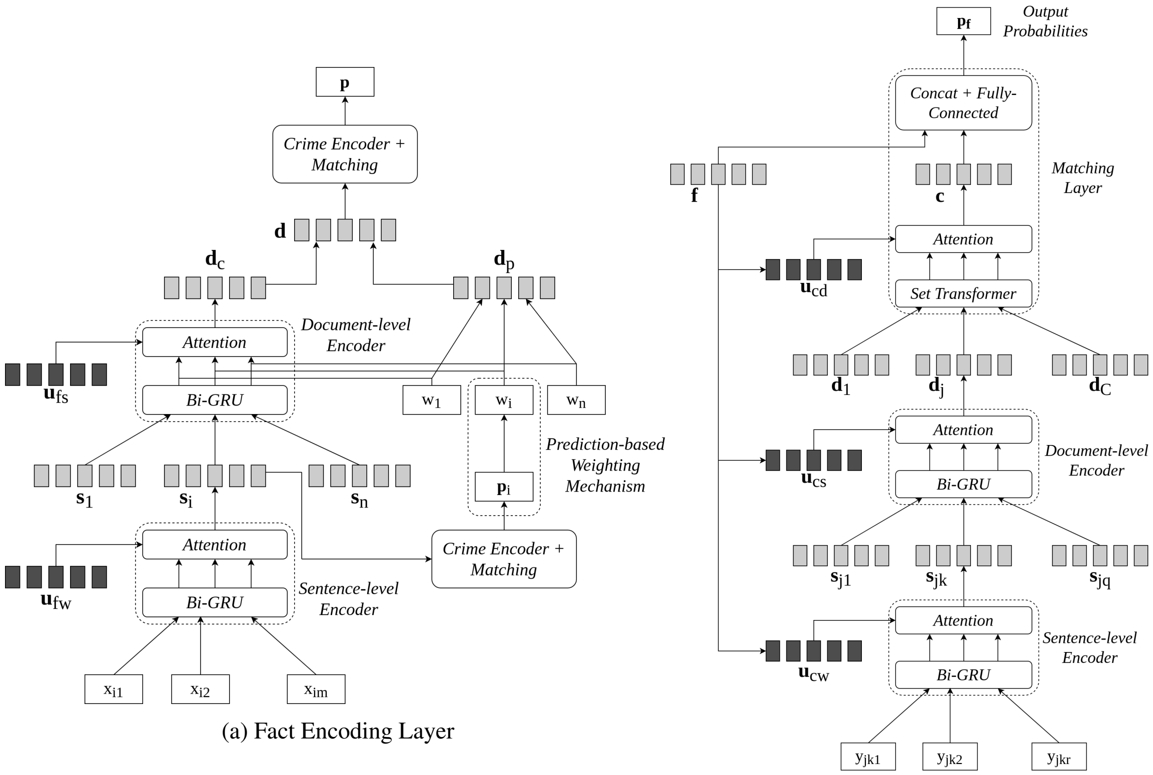
Fact Encoding Layer: We use a modified Hierarchical Attention Network (HAN) (Yang et al., 2016) to encode facts, as shown in Figure 1a. A HAN contains two Bi-GRU + Attention sequence encoders: a sentence-level encoder that produces a sentence embedding from words, and a document-level encoder that produces a document embedding from sentences. Thus, in our case, each fact sentence x_i is encoded to return a single embedding $\mathbf{s}_i$ for the entire sentence, using a trainable global context vector $\mathbf{u}_{fw}$ to calculate attention weights (Yang et al., 2016). The fact text, now a sequence of sentence embeddings $[\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n]$, is passed through the document-level encoder with a trainable global context vector $\mathbf{u}_{fs}$ to obtain an embedding for the entire document $\mathbf{d}_c$.

However, as discussed in Section 1, a lot of sentences in the fact descriptions are noisy (i.e., do not indicate any charge), and should be assigned low weights while constructing the document embedding. To guide the model towards distinguishing the noisy sentences, we pass the embedding s_i to the charge Encoder and Matching layer (detailed below) to obtain the sentence-level charge probabilities $\mathbf{p}_i = [p_{i1}, p_{i2}, \ldots, p_{iC}]$. We then learn a weight w_i for each sentence, from sentence-level predictions $\mathbf{p}_i$, and create another weighted document representation $\mathbf{d}_p$ as

$$w_i = \text{softmax}_i(\mathbf{W}^p \mathbf{p}_i + \mathbf{b}^p), \qquad \mathbf{d}_p = \sum_i w_i \mathbf{h}_i, \tag{1}$$

where $\mathbf{h}_i$ is the hidden embedding produced by document-level encoder Bi-GRU corresponding to sentence embedding $\mathbf{s}_i$. The final fact-side document embedding is obtained as $\mathbf{d} = \mathbf{d}_c + \mathbf{d}_p$.

Charge Encoding Layer: Another HAN is used to generate an embedding for each charge description. But, instead of using global context vectors to learn word and sentence attention weights, these weights are generated dynamically based on the given facts, similar to the approach used by Luo et al. (2017).



(a) Fact Encoding Layer

(b) Charge Encoding Layer and Matching Layer

Figure 1: Architecture of proposed model. The Fact Encoder (Figure 1a) generates fact-side sentence and document representations, which are independently passed to the Charge Encoder (Figure 1b) to generate charge-side representations, matching and producing predictions.

This helps the model to selectively attend to important sentences and words in the charge descriptions based on the given fact, and not important items in general.

As seen in Figure 1b, the input to this component is a fact-side representation $\mathbf{f}$, which can be one of the sentence embeddings $\mathbf{s}_i$, or the final document embedding $\mathbf{d}$. Using $\mathbf{f}$, we generate two context vectors $\mathbf{u}_{cw} = \mathbf{W}^w\mathbf{f} + \mathbf{b}^w$ and $\mathbf{u}_{cs} = \mathbf{W}^s\mathbf{f} + \mathbf{b}^s$, which are used for calculating attention weights for the two Bi-GRU + Attn encoders. Thereby, we generate charge-side document embedding $\mathbf{d}_j$ for each charge y_j.

Matching Layer: We now have a fact-side representation $\mathbf{f}$ and the dynamically generated set of charge-side document embeddings $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_C\}$. Similar to the charge Encoder, we learn another context vector: $\mathbf{u}_{cd} = \mathbf{W}^d\mathbf{f} + \mathbf{b}^d$. However, since the charge-side documents denote a set rather than a sequence, instead of using a Bi-GRU to model the correlations between the charges (as done by Luo et al. (2017)), we use a Set Transformer encoder as proposed by Lee et al. (2019). We use the Set Transformer encoder layer with two transformer stacks and 16 attention heads, and couple it with an attention layer with dynamically generated context $\mathbf{u}_{cd}$. This aggregates the charge-side document embeddings into a single embedding $\mathbf{c}$. The probabilities of charge classes for the given fact-side representation $\mathbf{f}$ are obtained by concatenating $\mathbf{f}$ and $\mathbf{c}$ and passing through a fully connected layer as

$$\mathbf{p_f} = \text{sigmoid}(\mathbf{W}^{class}[\mathbf{f}, \mathbf{c}] + \mathbf{b}^{class}), \tag{2}$$

where $\mathbf{W}^{class}$ and $\mathbf{b}^{class}$ represent the classification layer with C output units (C = no. of classes).

Prediction and Optimization: We thus obtain sentence-level charge probabilities $\mathbf{p}_i = [p_{i1}, p_{i2}, \ldots, p_{iC}]$ for each sentence x_i, and the document-level charge probabilities $\mathbf{p} = [p_1, p_2, \ldots, p_C]$. For every class, both at sentence and document level, we predict it to be relevant if the corresponding probability is greater than 0.5. We use weighted Binary Cross Entropy Loss for optimization using class weighting. The final loss is calculated by adding the sentence and document level losses.

charge	Frequency			charge	Frequency		
	Train-Doc	Train-Sent	Test-Doc		Train-Doc	Train-Sent	Test-Doc
Criminal Conspiracy (CC)	573	16	12	Theft (THE)	147	10	3
Offence against State (OS)	39	9	3	Robbery (ROB)	251	10	5
Unlawful Assembly (UA)	934	28	18	Mischief (MIS)	164	9	4
Offence against Public Justice (OPJ)	515	17	7	Cheating (CHE)	295	18	9
Offence affecting Public Safety (OPS)	62	5	3	Criminal Breach of Trust (CBT)	253	11	4
Offence related to Religion (OR)	12	5	3	Criminal Trespass (CT)	328	14	8
Murder (MUR)	2918	58	36	Forgery (FOR)	251	12	8
Hurt (HUR)	894	41	28	Marriage Offence (MO)	38	4	5
Kidnapping (KID)	267	11	4	Cruelty by Husband (CH)	316	13	5
Sexual Offence (SO)	296	17	8	Criminal Intimidation (CI)	193	7	5

Table 1: Document-level frequency of charges in each dataset. Train-Sent is a small subset of Train-Doc.

4 Dataset Preparation

We collected a large number of judgment documents from the Supreme Court of India containing facts of real legal situations. In this section, we describe the manner in which we extract information and build the datasets from the raw judgment documents.

4.1 Selecting Charges

The Indian legislature is divided into several documents, called Acts, which contain laws on a particular area. The Indian Penal Code, 1860 (IPC) is the Act which defines most charges. The IPC can be divided into the following descending order of hierarchies: (i) *Chapters*, containing charges of a similar nature, e.g., *Offences affecting Human Body*, (ii) *Topics*, indicating individual charges (e.g., *murder*, *hurt*), and (iii) *Sections*, which are laws regarding different aspects of a particular charge, viz., definition, punishment, and so on. We consider each Topic in IPC to be a charge which we wish to identify. Table 1 shows some examples of charges in the IPC. Each Topic contains one Section which broadly defines the charge; we consider the text of that Section to be the charge description. For instance, the charge *hurt* falls under the Chapter *Offences affecting Human Body*, and ranges from Section 319-338 of IPC. Section 319 actually defines the charge.

Indian judgment documents contain citations to Sections of IPC, which directly indicate which charges were committed in the given case. We use simple regular expression matching techniques to extract the Section citations from documents, and use the defined hierarchies in IPC to get the charges. We choose to consider the 20 most frequent charges (which are cited from most number of Supreme Court case documents) for our work, which are listed in Table 1.

4.2 Constructing Datasets

We selected documents which cite at least one of the 20 selected charges. This gives us a total of $4,338$ judgment documents with document-level charge information. Next, we extract the *facts of these cases*, to serve as descriptions of legal situations. Since Indian judgment documents are *not* clearly demarcated into the functional parts such as facts, precedents, judgment, etc., the facts need to be either extracted manually or using automated methods. We construct three datasets as described below:

Test-Doc: A set of 70 documents are chosen (so that they are distributed over all 20 selected charges), and the facts are manually extracted in consultation with legal experts (senior Law students from the Rajiv Gandhi School of Intellectual Property and Law, a reputed Law school in India).

Train-Doc: For the remaining $4,268$ documents, we employ an automated method developed in our prior work (Bhattacharya et al., 2019), which predicts the functional role for each sentence in Indian Supreme Court judgment documents, and extract those sentences that are predicted as *facts*. This method achieved an F1-score of 0.84 on identifying *facts*, so we consider it reliable for our purpose.

Train-Sent: From Train-Doc, we select a small subset of 120 documents (less than 3% of Train-Doc). As the document-level charges are already known (from the IPC Sections cited by each document), we consulted the legal experts to first extract the facts, and then identify the specific sentences indicating these charges. Every sentence is annotated with the possible charge that can be identified from that sentence alone. Most sentences (average 59%) in a fact description do not indicate any charge. For an example, see Table 4 later in the paper.

The documents in Test-Doc and Train-Sent are chosen in a manner to ensure similar distributions across all three datasets, as shown in Table 1. Note that Test-Doc contains only document-level charges, since this is the main task we are interested in. Only Train-Sent contains sentence-level charge annotations in addition to document-level charges. We had to restrict ourselves to only 120 and 70 documents in Train-Sent and Test-Doc respectively, since the manual annotation and fact extraction tasks need to be done by legal experts and hence it is expensive to scale these tasks to more documents.

5 Experiments and Results

This section describes our experiments and results. We start by describing the baseline models with which our proposed model is compared.

5.1 Baselines

We consider as baselines several domain-independent text-classification models as well as several domain-specific models built specifically for the task of identifying charges or legal articles.

Baselines that treat the textual descriptions as a flat sequence of words: We consider the following baselines – **(1) TF-IDF + SVM:** Features are extracted from the facts using TF-IDF vectorization (Salton and Buckley, 1988) and SVM classifier (Suykens and Vandewalle, 1999) is used to identify charges; **(2) Bi-GRU + Attn:** Bahdanau et al. (2014) demonstrated the use of a Bi-GRU to encode the sequence of words, and a trainable context vector to attentively select important words from the entire sequence; **(3) ACP:** Hu et al. (2018) selected 10 discriminative attributes to distinguish between confusing charges, and the charges and attributes are learned simultaneously in a multi-task learning framework; **(4) DPAM:** Wang et al. (2018) exploited the co-occurence of frequent articles with rare ones to learn better representations for the charge texts; **(5) HMN:** Wang et al. (2019) used the hierarchy between charges to learn better representations for rare charges from its sibling charges. In our work, we use the IPC Chapter information (see Section 4.1) as the parent labels.

Baselines that consider the texts as a hierarchy of sentences and words: We consider the following models that considers a hierarchical representation of texts (similar to our proposed model) – **(6) HAN:** Two stacks of Bi-GRU + Attention layers (Yang et al., 2016) are used to construct intermediate sentence embeddings as well as the final document embedding by selectively attending to important sentences and words; **(7) HAN-Siamese:** Utilizing the framework of González et al. (2019), we use two HANs to encode the facts and charge texts respectively, and a Siamese layer is placed on top to predict the probability of matching of each fact with each charge. **(8) Hier-BERT:** To overcome the length limitation of BERT (Devlin et al., 2018), Chalkidis et al. (2019) encode each sentence using BERT, and a Bi-GRU + Attn layer is used to construct the fact embedding from the sentence embeddings. **(9) FLA:** Luo et al. (2017) use two HANs to construct fact and charge text embeddings, and attentively select the important charges based on the facts. A Bi-GRU + Attn layer is used to aggregate the charge description embeddings before concatenating with fact embedding for classification.

Although none of these baselines utilize sentence-level charges, to make a fair comparison with our proposed model, we equip these models to utilize sentence-level information when training on Train-Sent. The hierarchical models generate intermediate sentence representations and thus can utilize multi-task learning (MTL) to model both sentence and document-level charges, when trained on Train-Sent. For obtaining sentence-level predictions, we follow the same technique as used for document-level predictions. The flat sequence models do not generate intermediate sentence embeddings and thus cannot utilize MTL. However, for the sake of fair comparison, we run these models on each sentence of a document independently, and obtain sentence-level predictions. The document-level predictions are generated by taking union of sentence-level predictions, uniformly for all models (to ensure fair comparison).

Fact Sequence Treatment	Model	Trained on Train-Doc			Trained on Train-Sent		
		Precision	Recall	F1	Precision	Recall	F1
Flat Sequence	TF-IDF + SVM	0.460	0.325	0.346	0.563	0.327	0.390
	Bi-GRU + Attn	0.092	0.643	0.145	0.158	0.697	0.192
	ACP	0.150	0.691	0.234	0.188	**0.705**	0.256
	DPAM	0.262	0.234	0.227	0.383	0.216	0.266
	HMN	**0.678**	0.450	**0.494**	**0.721**	0.451	0.521
Hierarchical Sequence	HAN	0.288	0.688	0.380	0.497	0.474	0.459
	HAN-Siamese	0.268	0.638	0.356	0.406	0.586	0.456
	Hier-BERT	0.180	**0.750**	0.267	0.519	0.555	0.515
	FLA	0.346	0.656	0.437	0.628	0.473	0.517
	Proposed Model	0.412	0.552	0.443	0.615	0.557	**0.566**

Table 2: Macro-averaged performance of the models on Test-Doc. Columns 3-5 show the performance when trained using only document-level charges from Train-Doc. Columns 6-8 show the performance when trained independently using both document and sentence-level charges from Train-Sent only. The differences in performance between our proposed model and the three closest baselines (HMN, FLA, Hier-BERT) are all statistically significant (paired t-Test with 95% confidence).

5.2 Experiment Settings

To understand the efficacy of using sentence-level charge information, we apply each model in two settings – (i) by training it on Train-Doc, and (ii) by training it *independently* on the small subset Train-Sent, where both sentence and document-level information are available. When using Train-Doc, it was divided into train and validation sets using a split ratio of $9 : 1$. This split was created randomly, and ensured to be same across all models. When using Train-Sent, we perform five-fold cross-validation. The performance of every model (irrespective of whether it was trained on Train-Doc or on Train-Sent) is measured over the Test-Doc set of 70 documents.

Pre-processing and Augmentations: We use the NER module of SpaCy (Honnibal and Montani, 2017) to identify and mask all the named entities with tokens like [PER] (person), [LOC] (location) and so on. We also use SpaCy to remove punctuations and stopwords, and split sentences. We use all the case documents at our disposal (except the Test-doc set) to pre-train Word2vec embeddings (Mikolov et al., 2013) for initializing the embedding layer of each model except Hier-BERT, which has a pretrained embedding layer. We also use a weighted Binary Cross Entropy Loss (as discussed in Section 3) for optimizing all the models, except DPAM, which has its custom loss function.

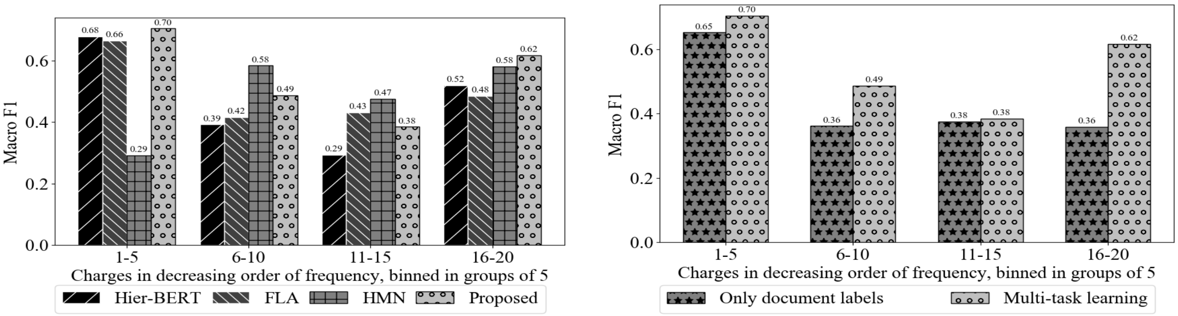
Hyper-parameters: For both facts and charge descriptions, embedding dimension was set to 128 for all models. All subsequent embeddings are also of the same dimension. Using batch size of 64 documents, all models were trained for 200 epochs. The model state at the best validation result was saved and used subsequently for testing on the Test-Doc set. Adam Optimizer (Kingma and Ba, 2014) with learning rate of 0.001 was used for all models, with L2 regularization of 0.0005 and dropout of 0.5.

Evaluation Metrics: All models generate document-level predictions irrespective of being trained on Train-Doc or Train-Sent. We compare the predictions of the models on Test-Doc with the gold standard labels, using macro-averaged Precision, Recall and F1-Score (macro average considered due to imbalanced nature of the dataset).

5.3 Results and Analysis

Table 2 shows the results of the two sets of experiments – training every model on Train-Doc and Train-Sent. All performances are measured over Test-Doc. In terms of macro-F1, our proposed model trained on Train-Sent achieves the best result (0.566 macro-F1).

Effects of using sentence-level charge labels: The advantage of having additional sentence-level charge information is clear from Table 2. All models show improvement in macro-F1, even when trained on the much smaller subset Train-sent (less than 3% of Train-Doc) having sentence-level charge labels. For the flat models, treating each sentence independently, and then pooling the sentence-level predictions to obtain document-level predictions leads to small improvements (average 15.4%) in macro-F1. For the hierarchical models, we see that Recall reduces, in general, when trained on Train-Sent, possibly due to



(a) Comparison of proposed model with closest baselines

(b) Comparison of proposed model with and without using multi-task learning

Figure 2: Performance analysis for frequent and rare charges. charges have been arranged in decreasing order of frequency and binned in groups of 5.

Component Removed	**Trained on Train-Sent**		
	P	**R**	**F1**
No pretrained embeddings	0.492	0.457	0.433
No weighted loss	0.626	0.394	0.436
No weightage based on sentence-level predictions	**0.627**	0.532	0.541
Using Bi-GRU + Attn instead of Set Transformer	0.545	0.533	0.529
All components used (last row of Table 2)	0.615	**0.557**	**0.566**

Table 3: Macro-averaged performance of the ablations on Test-Doc

the small size of Train-Sent. However, learning sentence-level charges greatly improves the Precision, thereby leading to high degree of improvement in F1 (average 37.6%). Especially for our proposed model, both Precision and Recall improves when trained on Train-Sent (as compared to Train-Doc).

Comparison of proposed model with baselines: The closest competitor, HMN, utilizes additional knowledge, i.e., the hierarchies between charges, and performs the best when considering only document-level charges. However, since it is a flat model, it cannot utilize Multi-Task Learning, and so the improvement is only marginal when trained on Train-Sent at sentence level. In contrast, our proposed model, though inferior to HMN when trained on Train-Doc (0.443 vs. 0.494 macro-F1), outperforms HMN when trained on Train-Sent, using MTL (0.566 vs. 0.521 macro-F1 – statistically significant difference).

Comparing our model with another competing baseline, FLA, we notice that there is small improvement (1.4%) when trained on Train-Doc, but the gap increases on Train-Sent (9.5%). This is because the prediction-based sentence weighting mechanism can work effectively only when the sentence-level loss is optimized. Finally, though Hier-BERT cannot compete with our model when trained on Train-Doc, it comes close when trained on Train-Sent, since the sentence-level loss helps the Hier-BERT model.

Performance on frequent and rare charges: To demonstrate the performance of various models on charges of different frequencies, we arrange the 20 selected charges in decreasing order of their frequencies, and group them into consecutive bins of five charges each. We calculate the macro-F1 over the charges in each bin.

Figure 2a compares the proposed model with the three closest competitors (HMN, FLA, Hier-BERT). The four groups of bars represent these four charge bins, with the most frequent charges' bin to the left and most rare charges' bin to the right. Our model outperforms Hier-BERT across all four bins, and FLA on all except the third bin. Since HMN utilizes hierarchy information, it outperforms our approach on those charges which are grouped together in the same Chapter of IPC, mainly occurring in the second and third bins. However, our model outperforms every baseline on the fourth bin (five most rare charges).

Figure 2b compares two versions of our proposed model, one trained on Train-Doc, and the other trained on Train-Sent using MTL. The version using MTL and sentence-level labels has higher performance across all bins. The highest improvement is for the fourth bin, again indicating that the MTL approach helps to identify the rare charges better.

Sentences (the facts of the case)	**Actual**	**Proposed**	**HMN**	**FLA**	**Hier-BERT**
1. It has been stated that A-1 to A-7 demanded that the appellant would vacate the above-mentioned house.					
2. They tried to evict the appellant forcibly from the house but locality of the people intervened and made their efforts futile.			CI	CI	
3. However, on 10.06.1990 at 7.00 AM, [PER] (PW-2) while taking water from a tap which was near the gate saw A-1 to A-7 entering the premises.	UA	UA		UA	
4. She went and informed her mother [PER] (PW-7).					
5. The appellant (PW-1) and PW-7 closed the doors of the house so as to prevent the entry of A-1 to A-7 and their associates.					
6. However, A-1 to 7 broke open the doors of the house and gained entry into the house.	UA, CT	CT	CT	CT	CT
7. When they attempted to lift the household articles, the appellant and his sister's husband [PER] went through another door of the house to the police station to inform the highhanded acts of the accused.	ROB		ROB		
8. When PWs 7, 9, 10 and 13 prevented A-1 and his associates from removing the household articles, they were beaten up by the accused.	HUR	HUR	HUR	HUR	HUR
9. By the time appellant returned from the police station, A-1 and his associates loaded household articles in a lorry and emptied the house.	ROB	ROB	ROB	ROB	ROB
Overall (Document-level)	UA, CT, ROB, HUR	UA, CT, ROB, HUR	CI, CT, ROB, HUR	CI, UA, CT, ROB, HUR	CT, ROB, HUR

Table 4: Predictions of our proposed model and baselines for a sample case, along with actual labels (as identified by legal experts). Both sentence-level and document-level labels shown. Here UA: *unlawful assembly*, CT: *criminal trespass*, ROB: *robbery*, HUR: *hurt*, and CI: *criminal intimidation*.

Ablation tests: We investigate the effects of our architectural contributions and augmentations by ablation tests (results in Table 3). We observe that using pretrained Word2vec embeddings and class weights during loss calculation has profound impact on performance, since the Macro-F1 dips sharply (by 24.6% for both) when removing either of these components. Secondly, if the sentence weights are calculated only on the basis of the learned global context vector (and not via sentence-level predictions), the fact encoder becomes a vanilla HAN. The performance decreases slightly (by 4.4%), indicating that better document representations can be learnt by additional weightage on the sentence-level predictions. Finally, we replace the Set Transformer Encoder with a standard Bi-GRU layer, similar to FLA model to aggregate the 20 charge text embeddings. The performance decreases slightly (by 6.5%), indicating that the order-invariant Set Transformer is a better method of aggregating charge texts than Bi-GRU + Attn, which is order-sensitive (Luo et al., 2017).

5.4 A case study

To demonstrate the effectiveness of our approach, and the utility of sentence-level predictions towards increasing explainability of charge predictions, we show the predictions of the best four performing models (all trained on Train-Sent) on one example fact from Test-Doc (for which we explicitly obtain actual sentence-level annotations from the legal experts). As shown in Table 4, the actual charges for this example are *unlawful assembly* (UA), *criminal trespass* (CT), *robbery* (ROB) and *hurt* (HUR). Out of 9 sentences, the sentences numbered 1, 2, 4 and 5 do not indicate any charge (noisy sentences).

At the document-level, our proposed model is able to predict all four charges correctly; however, it is unable to identify UA for Sentence 6 and ROB for Sentence 7. HMN is unable to identify UA at all, probably because UA does not have any siblings in the IPC hierarchy, thus making it difficult for HMN. Also, HMN wrongly predicts that Sentence 2 indicates *criminal intimidation* (CI). Although the predictions of FLA are almost same as our proposed model, it also wrongly predicts Sentence 2 as CI. Hier-BERT is unable to predict UA but it does not wrongly identify any charge.

From the point of view of explainability, we can use the sentence-level predictions to explain the document-level outputs of a model. For instance, we can say that UA is applicable due to Sentence 3, and CT is applicable due to Sentence 6 (for our proposed model). Also the sentences that lead to wrong charges being identified can be pointed out as well. Thus, sentence-level charge identification not only helps to improve document-level performance, but also to generate sound explanations.

6 Conclusion

In this paper, we show that using a small set ($< 3\%$) of documents with sentence-level crime labels with a multi-task learning setup (to predict sentence and document-level crimes) improves overall ACI performance across a range of models, as compared to training the models with only document level crime information on a much larger dataset. We also propose a model that makes use of multi-task learning and a novel weighting scheme to learn better document representations by distinguishing between crime-indicative and noisy sentences. The implementation of our proposed model is available at `https://github.com/Law-AI/automatic-charge-identification`.

In future, we plan to explore several directions. For instance, better representations can be learnt for rare crimes by utilizing co-occurrence signals (Wang et al., 2018) and existing hierarchies between crimes (Wang et al., 2019). Also, we can try to predict the relevant statutes (Sections in IPC) directly instead of the crimes (Topics in IPC) as we have done in this paper.

Ethics Statement

Since the task of charge prediction has a critical real-world application, there are some ethical issues worth discussing.

First, what are the real world use-case scenarios for such a model? We wish to emphasize that this model is *not* designed to be directly employed in the litigation process by replacing the humans involved in the process (such as police personnel and law practitioners). Rather, the model is meant to provide a recommendation to the humans to speed up the litigation process, which itself can be highly beneficial in countries with an overburdened legal system, such as India. It can also be used as a tool to review the charges already identified by humans. The law experts whom we consulted for annotation of the data were made aware of the intended use-case, and they agreed that such models/tools can be of great help in the Indian legal scenario. The dataset was formulated in close consultation with the law experts.

Second, can anonymization of data (or lack thereof) be an issue? Ideally, the charges should only be predicted based on the events in the fact description, and independent of the named entities (e.g., the parties involved). However, Indian Supreme Court judgment documents are publicly available in non-anonymized form (e.g., at `https://main.sci.gov.in/judgments`), and the names of the parties are mentioned throughout the text. As stated earlier in the paper, we have used NER masking to preprocess the data, in order to prevent any biases resulting from the names of the involved parties.

It can also be noted that "charges" are formally identified only at the document-level, i.e., after going through the entire facts of the situation. However, we associate the "charge" label with each sentence, as a rough estimate of whether the particular sentence possibly indicates one or more of these charges. This is what we actually mean by "sentence-level charges" in this work.

Finally, automatic charge prediction via neural methods is still an emerging technology, and these models need to be equipped with explainability systems to make the model outputs easy to comprehend even for non-domain users. However, in the current form, the model outputs can only be interpreted correctly by legal experts, and hence it is advisable to use this model only for recommendation purposes.

Acknowledgements

The authors thank the Law students who helped in developing the gold standard data. The research is partially supported by SERB, Government of India, through a project titled "NYAYA: A Legal Assistance System for Legal Experts and the Common Man in India".

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Paheli Bhattacharya, Shounak Paul, Kripabandhu Ghosh, Saptarshi Ghosh, and Adam Wyner. 2019. Identification of rhetorical roles of sentences in indian legal judgments. In *Proceedings of JURIX*.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in english. In *Proceedings of ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

J-A González, L-F Hurtado, E Segarra, Fernando García-Granada, and E Sanchis. 2019. Summarization of spanish talk shows with siamese hierarchical attention networks. *Applied Sciences*, 9(18):3836.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Zikun Hu, Xiang Li, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2018. Few-shot charge prediction with discriminative legal attributes. In *Proceedings of COLING*, pages 487–498.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Fred Kort. 1957. Predicting supreme court decisions mathematically: A quantitative analysis of the "right to counsel" cases. *American Political Science Review*, 51(1):1–12.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of ICML*, pages 3744–3753.

Wan-Chen Lin, Tsung-Ting Kuo, Tung-Jia Chang, Chueh-An Yen, Chao-Ju Chen, and Shou-de Lin. 2012. Exploiting machine learning models for chinese legal documents labeling, case classification, and sentencing prediction. *Proceedings of ROCLING*, page 140.

Chao-Lin Liu and Chwen-Dar Hsieh. 2006. Exploring phrase-based classification of judicial documents for criminal charges in chinese. In *International Symposium on Methodologies for Intelligent Systems*, pages 681–690. Springer.

Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to predict charges for criminal cases with legal basis. In *Proceedings of EMNLP*, pages 2727–2736.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Jeffrey A Segal. 1984. Predicting supreme court cases probabilistically: The search and seizure cases, 1962-1981. *American Political Science Review*, 78(4):891–900.

Johan AK Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.

S Sidney Ulmer. 1963. Quantitative analysis of judicial processes: Some practical and theoretical applications. *Law & Contemp. Probs.*, 28:164.

Pengfei Wang, Ze Yang, Shuzi Niu, Yongfeng Zhang, Lei Zhang, and ShaoZhang Niu. 2018. Modeling dynamic pairwise attention for crime classification over legal articles. In *Proceedings of SIGIR*, pages 485–494. ACM.

Pengfei Wang, Yu Fan, Shuzi Niu, Ze Yang, Yongfeng Zhang, and Jiafeng Guo. 2019. Hierarchical matching network for crime classification. In *Proceedings of SIGIR*, pages 325–334. ACM.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. Cail2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*.

Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish confusing law articles for legal judgment prediction. In *Proceedings of ACL*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL*, pages 1480–1489.

Wenmian Yang, Weijia Jia, XIaojie Zhou, and Yutao Luo. 2019. Legal judgment prediction via multi-perspective bi-feedback network. In *Proceedings of IJCAI*.

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of EMNLP*, pages 3540–3549.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does nlp benefit legal system: A summary of legal artificial intelligence. In *Proceedings of ACL*.