Legal Argument Extraction from Court Judgements using Integer Linear Programming

Basit Ali[1], **Sachin Pawar**[1], **Girish K. Palshikar**[1], **Anindita Sinha Banerjee**[1]
Dhirendra Singh[2]
[1]TCS Research, Tata Consultancy Services Limited, India.
[2]CFILT, Indian Institute of Technology Bombay, India.
{ali.basit, sachin7.p, gk.palshikar, anindita.sinha2}@tcs.com
dhirendra.singh@iitb.ac.in

Abstract

Legal arguments are one of the key aspects of legal knowledge which are expressed in various ways in the unstructured text of court judgements. A large database of past legal arguments can be created by extracting arguments from court judgements, categorizing them, and storing them in a structured format. Such a database would be useful for suggesting suitable arguments for any new case. In this paper, we focus on extracting arguments from Indian Supreme Court judgements using minimal supervision. We first identify a set of certain sentence-level *argument markers* which are useful for argument extraction such as whether a sentence contains a *claim* or not, whether a sentence is argumentative in nature, whether two sentences are part of the same argument, etc. We then model the legal argument extraction problem as a text segmentation problem where we combine multiple *weak evidences* in the form of argument markers using Integer Linear Programming (ILP), finally arriving at a global document-level solution giving the most optimal legal arguments. We demonstrate the effectiveness of our technique by comparing it against several competent baselines.

1 Introduction

In the field of argument mining, extraction of legal arguments from court judgements has been receiving increasing attention (Poudyal et al., 2020; Grundler et al., 2022; Habernal et al., 2023). Most of these approaches are supervised in nature in the sense that they need a significantly large corpus of documents from a specific area (e.g., ECHR - European Court of Human Rights) which are annotated with legal arguments. In this paper, we focus on extracting legal arguments from Indian Supreme Court judgements using minimal supervision. Our goal is to construct a large database of past legal arguments by extracting legal arguments from court judgements, categorizing them, and storing them in a structured format. Such a database would be useful in building a high-level legal decision support system where some of its features could be – i) suggesting suitable arguments given a new case description, ii) learning to estimate the strength of a new argument based on the similar past arguments that helped to win the case.

In this paper, we focus specifically on extraction of legal arguments and to the best of our knowledge, this is the first such attempt for – i) legal argument extraction without any in-domain supervision and ii) argument extraction from Indian court judgements. For categorizing the arguments, we propose to simply map them to the *statute facets* which were recently proposed in our previous work (Pawar et al., 2023). A statute facet is any important specific aspect of an Act which can be potentially used in legal arguments in a case related to the Act. For example, following are statute facets from India's Industrial Disputes Act – *workman*, *illegal strikes*, and *notice of retrenchment*.

We consider a *legal argument* as *a sequence of contiguous sentences in a court judgement which constitute a complete and coherent argument.* A legal argument generally consists of a sentence containing a major *claim* (or conclusion) and other sentences acting as sufficient *premises* for that claim. Table 1 shows a few examples of such legal arguments where the statute facets from India's Industrial Disputes Act (1947) are also underlined.

A major challenge in legal argument extraction from Indian court judgements is the unavailability of a training dataset where the legal arguments are annotated by human experts. Hence, we first propose to identify certain *argument markers* within sentences of a court judgement which are *weak* indicators of presence of a legal argument. Here, we refer to these argument markers as weak evidences because individually any marker is not a strong enough indicator of a legal argument and it is also not possible to automatically identify these

Proceedings of the 10th Workshop on Argument Mining, pages 52–63

Arguments
- There were different systems of dearness allowance for the operators and the clerical and subordinate staff in the appellant company. *- That such a different system of dearness allowance for employees working under the same employer is not warranted is clear from the decisions of this Court in the cases of Greaves Cotton & Co. and Bengal Chemical & Pharmaceutical Works Ltd.* *- Therefore the Tribunal was justified in devising a uniform scale of dearness allowance applicable to all the employees of the appellant.* (**claim**)
- It is therefore clear that the claim for bonus can be made by the employees only if as a result of the joint contribution of capital and labour the industrial concern has earned profits. (**claim**) *- If in any particular year the working of the industrial concern has resulted in loss there is no basis nor justification for a demand for bonus.* *- Bonus is not a deferred wage, because if it were so it would necessarily rank for precedence before dividends.* *- The dividends can only be paid out of profits and unless and until profits are made no occasion or question can also arise for distribution of any sum as bonus amongst the employees.* *- If the industrial concern has resulted in a trading loss, there would be no profits of the particular year available for distribution of dividends, much less could the employees claim the distribution of bonus during that year.*

Table 1: Examples of legal arguments from court judgements related to Industrial Disputes Act.

Argument Marker	What does it indicate for a sentence S?
Claim sentence (C)	whether S makes any claim or draw some conclusion
Argumentative sentence (A)	whether S is argumentative in nature, i.e., is it either a claim or a premise of some argument
Sentence pair relation (SP)	whether S and its previous sentence belong to the same argument
Statute Facets (F)	the statute facets mentioned in S
Discourse connectors (D)	whether S has a discourse relation with its previous sentence through a causal discourse marker such as *therefore* or *hence*
Argument agent (AA)	whether S has a different argument agent (i.e., entity making the argument) than its previous sentence
Subjectivity score (SS)	whether S is a subjective sentence

Table 2: List of various argument markers used

argument markers with high accuracy. Table 2 shows the list of various argument markers used and it can be observed that the statute facets are also used as one of the argument markers. Each argument marker is identified either by using linguistic rules/patterns (for C, F, D, AA) or, by learning a classifier using training data from another area – ECHR (for AS and SP), or by using an off-the-shelf library (for SS). We then use Integer Linear Programming (ILP) to combine the weak evidences provided by these argument markers to arrive at a final document-level solution leading to identification of legal arguments. The ILP framework also enables us to represent various domain rules in the form of constraints and objectives. The main contributions of this work are:

• **Argument markers**: Techniques for identifying various argument markers (Section 3.1).

• **ArgExt-ILP**: An ILP-based technique for legal argument extraction (Section 3.3).

• **Dataset**: A dataset of 10 court judgements from Indian Supreme Court containing 127 arguments, which is the first such arguments-annotated dataset for Indian court judgements (Section 5.1).

• **Evaluation metrics**: A set of evaluation metrics for comparing the predicted arguments with the gold-standard arguments (Section 5.3).

2 Problem Definition

The problem is formally defined as follows:

Input: (i) A court judgement document J (sequence of N sentences $S_1, S_2, \cdots, S_N$), and (ii) A set of statute facets $f_1, f_2, \cdots, f_k$ for an Act A

Output: A set of extracted arguments where any i^{th} argument is a tuple (i_s, i_e) such that all the contiguous sentences starting from S_{i_s} to S_{i_e} constitute the argument.

Scope and assumptions: If there are multiple arguments present in J, they must be mutually exclusive, i.e., no sentence is common between any two such arguments. Also, another simplifying assumption is that an argument consists of contiguous sentences which may not be always true[1]. Extending our techniques to extract non-contiguous arguments is to be tackled as a part of future work.

3 Proposed Techniques

In this section, we describe identification of various argument markers and our proposed argument extraction techniques which use these markers.

[1] In ECHR corpus (Poudyal et al., 2020), almost 50% arguments consist of contiguous text

3.1 Argument Markers

3.1.1 Claim sentences (C)

As any legal argument must contain at least one claim sentence, it becomes one of the most important argument markers. It is very challenging to identify claim sentences without any direct supervision. We attempted to train sentence classifiers to identify claims using training data from ECHR corpus as well as using zero-shot text classification using open source LLMs like falcon-7b-instruct (Almazrouei et al., 2023). However, these attempts were not successful. Therefore, we designed a set of linguistic rules/patterns by observing the claim sentences in court judgements.

LR1: If a sentence contains a copula verb which is modified by a causal discourse marker (e.g., *therefore*, *hence*) as an adverbial modifier then it may be a claim. E.g., *Therefore, he was not a workman*.

LR2: If a sentence contains a non-copula verb which is modified by a causal discourse marker as an adverbial modifier and also modified by a modal verb (e.g., *would*, *could*) then it may be a claim. E.g., *Therefore, as Ram was not a workman his case would not be covered by the IDA...*

LR3: We prepared a list[2] of nouns and verbs which indicate some kind of claim, conclusion, view, or opinion. Examples of such nouns/verbs are *opinion*, *conclusion*, *contended*, *concluded*, etc. If a sentence contains any of these followed by a complement clause containing actual claim/conclusion/opinion then it may be a claim. For example, consider the following sentences where such noun/verb and the complement clause are highlighted – *We are of the opinion that the High Court erred in not awarding compensation to the appellant.*, *The learned counsel contended that the respondent was denied a fair hearing*.

LR4: We also prepared a list of adjectives and adverbs with positive or negative sentiment, e.g., *erroneous*, *incorrectly*, *valid*, *wrongly*, *illegally*. If a sentence contains any one of these words to evaluate something or to express an opinion about something, then it may be a claim. Following are example sentences – *The order of the Labour Court deciding the reference against the respondent-workman is illegal.*, *The said stand of the workers union is not consistent with the nature of the complaint*.

[2]The complete lists of words used in these patterns are provided in Appendix A.

3.1.2 Argumentative sentences (AS)

Identification of argumentative sentences has been studied in many domains (e.g., essays, debates, legal, etc.) and the techniques employed are mostly supervised in nature (Poudyal et al., 2020). Argumentative sentences can be thought of as a superset of claim sentences in the sense that both claims as well as their premises are part of argumentative sentences. We used a BERT-based sentence classifier which combines the [CLS] representation of a sentence and attention weighted average of the other tokens to get the overall representation of the sentence. It is trained using training data from multiple sources (e.g., ECHR corpus, essay corpus, rhetorical role corpus, and Indian judgements corpus) as described in Ali et al. (2022).

3.1.3 Sentence pair relation (SP)

The goal here is to predict whether any two sentences belong to the same argument or not. For this, we used a BERT-based sentence pair classifier (where two sentences are separated by a [SEP] token) which is trained using the ECHR corpus (Poudyal et al., 2020). The positive training examples (10418) are created by taking all the pairs present within an argument whereas the equal number of negative pairs are chosen randomly such that the sentences in each pair are not part of the same argument. We used this classifier for each pair of contiguous sentences in a court judgement to predict the probability that these sentences belong to the same argument.

3.1.4 Statute facets (F)

We considered all the noun phrase facets extracted from Industrial Disputes Act[3] using the technique described in previous work (Pawar et al., 2023). We matched each facet with each sentence in a court judgement ensuring that morphological variations are handled (e.g., *employer* and *employers*). The intuition is that if a facet is present in a sentence then it is more likely to be argumentative in nature. Moreover, presence of a common facet across most sentences in an argument is also a weak measure of coherence. E.g., in the first argument of Table 1, the facet *dearness allowance* is present in all its sentences. Hence, even though *statute facets* are not strong indicators of a legal argument on their own, they may help as weak argument markers (see ablation results in Section 5.4).

[3]Because all our test files are chosen to be related to IDA.

3.1.5 Discourse connectors (D)

If a sentence is connected with its previous sentence through a causal discourse connector (e.g., *therefore*, *consequently*) then it is a strong indication of coherence between the two sentences. Moreover, it is also a weak indication of the current sentence being a claim. Hence, we identify this information about discourse connectors using the rules described in Ali et al. (2022).

3.1.6 Argument agent (AA)

An argument agent is the entity who is putting forward any argument such as *appellant*, *lower court*, or *respondent*. If argument agents of the two contiguous sentences are different then it is a good indicator of non-cohesion between them. Hence, for each sentence, we identify whether its argument agent is different from its previous sentence using the rules described in Ali et al. (2022).

3.1.7 Subjectivity score (SS)

We compute subjectivity score for each sentence in a court judgement using TextBlob library[4]. Here, the intuition is that if a sentence is subjective then it is more likely to be an opinion or a claim.

3.2 ArgExt-Rules

We propose a simple rule-based technique which uses the information about argument markers in a court judgement to extract legal arguments from it. Algorithm 1 describes this technique in detail. Intuitively, this technique simply tries to extract a set of coherent and complete arguments without using any optimization technique, ensuring that either the first or last sentence in each argument is a claim sentence along with some additional constraints. It expands each claim sentence (say S_i for which $C[i] = 1$) in either forward or backward direction to identify a complete argument. While expanding the argument in either of the directions, it adds a new sentence to the argument only if that sentence mentions at least one facet from F and it lies in the same paragraph as that of S_i. A new sentence may still be added even if it does not mention any facet but at most one such sentence is allowed in an argument only as an intermediate sentence. As S_i is expanded in both forward and backward directions, the above process results in two candidate arguments – R_1 (where S_i is expanded backward) and R_2 (where S_i is expanded forward), where only one of them has to be selected. If S_i contains a discourse marker which connects it to its previous sentence (i.e., if $D[i] = 1$) then R_1 is selected as a more coherent argument. Otherwise, average sentence pair similarity score is computed for both R_1 and R_2 and the one with higher score is selected. The algorithm may result in overlapping arguments which are resolved as follows. For each pair of overlapping arguments, we discard that argument which contains lesser number of argumentative sentences than the other.

Data: J: court judgement with N sentences $\{S_1, \cdots S_N\}$, C: binary array of length N s.t. $C[i] = 1$ if i^{th} sentence contains a claim, P: array of length N s.t. $P[i]$ indicates paragraph number, D: binary array of length N s.t. $D[i] = 1$ if i^{th} sentence is connected to its previous sentence through a causal discourse marker, SP: real-valued array of length N s.t. $SP[i]$ indicates the probability that i^{th} and $(i-1)^{th}$ sentences are part of the same argument, F: set of statute facets from act A
Result: $Args$: set of arguments extracted from J

```
Args := {}
for S_i ∈ J do
    if C[i] == 1 then
        R_1 := {S_i}; j := i − 1
        while S_j exists AND S_j contains at least
          one facet from F AND P_j == P_i do
            R_1 := R_1 ∪ {S_j}; j := j − 1
        R_2 := {S_i}; j := i + 1
        while S_j exists AND S_j contains at least
          one facet from F AND P_j == P_i do
            R_2 := R_2 ∪ {S_j}; j := j + 1
        if D[i] == 1 then Args := Args ∪ R_1 ;
        else
            PR_1 := Avg pairwise SP values in R_1
            PR_2 := Avg pairwise SP values in R_2
            if PR_1 > PR_2 then
              Args := Args ∪ R_1 ;
            else Args := Args ∪ R_2 ;
return Args
```

Algorithm 1: Algorithm for ArgExt-Rules

3.3 ArgExt-ILP

We now describe our principal technique ArgExt-ILP which uses Integer Linear Programming (ILP) for combining multiple weak evidences provided by argument markers to extract actual arguments. ILP provides a suitable framework where the constraints and the objective can incorporate – (i) the information about argument markers (e.g., *each argument should start or end with a claim sentence*) and (ii) various types of domain knowledge about legal arguments (e.g., *an argument is unlikely to cross paragraph boundaries*). Thus, an optimal solution to an ILP program leads to a set of predicted arguments which conform to the argument markers

[4] https://textblob.readthedocs.io/en/dev/

and satisfy these domain rules as much as possible.

Tables 3 and 4 show our ILP formulation in detail. For each input document (i.e., court judgement J), an ILP program is prepared using the information about various argument markers in that document. The ILP program is then solved to obtain the predicted arguments from that document. The information about argument markers is provided to ILP through various input parameters such as C (claim sentences), AS (argumentative sentences), SP (sentence pair relations) as described in Table 3. The decision variables X and Y are binary variables. They are designed to represent the output (i.e., the predicted arguments) in such a way that the j^{th} column of the matrix $X - Y$ contains 1's in only those rows which correspond to sentences constituting the j^{th} argument (see Table 3). In other words, $(X[i,j] - Y[i,j])$ equals 1 if and only if i^{th} sentence is part of the j^{th} argument. The constraints C_1 to C_5 ensure that the extracted arguments are non-overlapping and correspond to contiguous sentences only. The constraint C_6 ensures that each extracted argument contains a claim sentence as its first or last sentence. For any j, $(X[i,j] - X[i-1,j])$ is 1 for only one i (because of the constraint C_1) which corresponds to the first sentence of the j^{th} argument. Similarly, for any j, $(Y[i+1,j] - Y[i,j])$ is 1 only for one i (because of the constraint C_2) which corresponds to the last sentence of the j^{th} argument. Hence, the left hand side of C_6 is at least 1 if and only if j^{th} argument contains a claim sentence as its first or last sentence. Also, the right hand side of C_6, i.e., $X[N,j]$ is 1 only if j^{th} argument exists, otherwise it is 0. Similarly, other constraints C_7 to C_9 are added to conform to other domain knowledge based rules as described in Table 3. Table 4 describes the objective which is minimized. The overall objective consists of 3 terms. The first term Obj_1 attempts to minimize the number of claim, subjective, and argumentative sentences which are not part of any extracted argument. Obj_2 ensures that as far as possible, the sentence pairs on argument boundaries are not related to each other. Obj_3 tries to maximize the overall number of facets which are part of the extracted arguments.

4 Related Work

Extraction of legal arguments: We discuss some of the most relevant techniques for extraction of legal arguments here. Poudyal (2016) identified the argumentative sentences and used soft clustering technique to form an argument which consists of premises and claims. They automatically identified the premise/claim structure within an argument using multiple features such as lexical, syntactic (tree kernel), dependency, n-gram, etc. The top n features are selected using gain-ratio for both classifying argumentative and premise/claim type sentences. Wei et al. (2017) proposed to use ILP to jointly solve multiple sub-tasks in argument mining such as argumentation component type classification and relation classification. We are also using ILP in our proposed technique, but we have modelled argument extraction differently as a text segmentation problem. One of the most significant work in legal argument extraction is by Poudyal et al. (2020) where they released an arguments-annotated corpus of 42 judgements of European Court of Human Rights (ECHR). They also presented BERT-based baseline techniques for three key tasks in argument extraction – argument clause recognition, clause relation prediction, and premise/conclusion recognition. Grundler et al. (2022) released *Demosthenes* which is a corpus of 40 judgements of the Court of Justice of the European Union on matters of fiscal state aid. The corpus contains annotations for three hierarchical levels of information – the argumentative elements, their types, and their argument schemes. Recently, Habernal et al. (2023) proposed an interesting alternate perspective that rather than simplifying arguments into generic premises and claims, it is more important to capture rich typology of arguments for gaining insights into the particular case and applications of law in general. They proposed a new annotation scheme accordingly for capturing 16 argument types and 5 argument actors for each argument, where an argument is a text span. The text span of an argument was allowed to cross sentence boundaries but not paragraph boundaries. They released a large corpus of 373 annotated court decisions and also proposed sequence labelling techniques for identifying argument text spans. We have used their model as one of the baselines. Other techniques for legal argument extraction are (Mochales and Moens, 2011; Trautmann, 2020; Xu and Ashley, 2023; Zhang et al., 2023; Santin et al., 2023).

Text Segmentation: This task is relevant for our work because we have modelled argument extraction as a text segmentation problem. Some generic

Input parameters:
N: No. of sentences in the court judgement J
M: Maximum no. of arguments in any court judgement
K: Total no. of facets in the Act A
C: Binary array of length N such that $C[i] = 1$ iff i^{th} sentence contains any *claim*. (Section 3.1.1)
D: Binary array of length N such that $D[i] = 1$ iff i^{th} sentence contains support indicating discourse markers such as *therefore* and *consequently* which link it to the $(i-1)^{th}$ sentence. (Section 3.1.5)
AS: Binary array of length N such that $AS[i] = 1$ iff i^{th} sentence is argumentative in nature. (Section 3.1.2)
AA: Binary array of length N such that $AA[i] = 1$; iff i^{th} sentence's argument agent (such as *appellant*, *respondent*, *lower court*, *judge*) is different from the previous sentence's agent. (Section 3.1.6)
F: Binary matrix of size $N \times K$ such that $F[i,k] = 1$ iff i^{th} sentence contains k^{th} facet and $F[i,k] = 0$ otherwise (Section 3.1.4)
P: Binary array of length N such that $P[i] = 1$; iff i^{th} sentence belongs to a new (different) paragraph as compared to the $(i-1)^{th}$ sentence.
SP: Real-valued array of length N such that $SP[i] =$ the probability that the i^{th} sentence and the $(i-1)^{th}$ sentence belong to the same argument. (Section 3.1.3)
SS: Real-valued array of length N such that $SS[i] =$ the subjectivity score of the i^{th} sentence. (Section 3.1.7)

Decision variables:
X: Binary matrix of size $N \times M$ such that $X[i,j] = 1, \forall_{i \geq k}$ iff j^{th} argument starts at the k^{th} sentence. $X[i,j] = 0, \forall_{i<k}$
Y: Binary matrix of size $N \times M$ such that $Y[i,j] = 1, \forall_{i>k}$ iff j^{th} argument ends at the k^{th} sentence. $Y[i,j] = 0, \forall_{i \leq k}$

Constraints:
C_1: For a fixed j, $X[:,j]$ should be monotonically increasing. $\quad X[i-1,j] \leq X[i,j];\ \forall_{i,j}\ s.t.\ 2 \leq i \leq N, 1 \leq j \leq M$

C_2: For a fixed j, $Y[:,j]$ should be monotonically increasing. $\quad Y[i-1,j] \leq Y[i,j];\ \forall_{i,j}\ s.t.\ 2 \leq i \leq N, 1 \leq j \leq M$

C_3: The start of an argument should be before its end. $\quad X[i,j] \geq Y[i,j];\ \forall_{i,j}\ s.t.\ 1 \leq i \leq N, 1 \leq j \leq M$

C_4: j^{th} argument should start only after $(j-1)^{th}$ argument ends. $\quad Y[i,j-1] \geq X[i,j];\ \forall_{i,j}\ s.t.\ 1 \leq i \leq N, 2 \leq j \leq M$

C_5: Any argument should contain at least one sentence.
$\sum_{i=1}^{N-1}((i+1) \cdot (Y[i+1,j] - Y[i,j])) - \sum_{i=2}^{N}(i \cdot (X[i,j] - X[i-1,j])) \geq X[N,j];\ \forall_j\ s.t.\ 1 \leq j \leq M$

C_6: At least one of the first sentence or the last sentence of any argument should contain a claim.
$\sum_{i=2}^{N}(C[i] \cdot (X[i,j] - X[i-1,j])) + \sum_{i=1}^{N-1}(C[i] \cdot (Y[i+1,j] - Y[i,j])) \geq X[N,j];\ \forall_j\ s.t.\ 1 \leq j \leq M$

C_7: Any argument should not start with a sentence containing discourse connector to its previous sentence.
$\sum_{i=2}^{N} D[i] \cdot (X[i,j] - X[i-1,j]) \leq 0;\ \forall_j\ s.t.\ 1 \leq j \leq M$

C_8: If a sentence contains an argument agent which is different from the previous sentence then such sentence can either be the first sentence in some argument or it may not be part of any argument.
$\sum_{j=1}^{M}(X[i,j] - Y[i,j]) - \sum_{j=1}^{M}(X[i,j] - X[i-1,j]) + AA[i] \leq 1;\ \forall_i\ s.t.\ 2 \leq i \leq M$

C_9: Any argument should not be spread across multiple paragraphs.
$(X[i,j] - Y[i,j]) - (X[i,j] - X[i-1,j]) + P[i] \leq 1;\ \forall_{i,j}\ s.t.\ 2 \leq i \leq N, 1 \leq j \leq M$

Table 3: Input parameters, decision variables and constraints used in ArgExt-ILP

Objective: Minimize $Obj_1 + Obj_2 - Obj_3$

Obj_1: Minimize the number of claim, argumentative, and subjective sentences which are not part of any extracted argument.
$Obj_1 = \sum_{i=1}^{N}(C[i] + SS[i] + AS[i]) \cdot \left(1 - \left(\sum_{j=1}^{M}(X[i,j] - Y[i,j])\right)\right)$

Obj_2: Minimize the average of probability scores that i^{th} and $(i-1)^{th}$ sentences belong to the same argument when they occur on an argument boundary.
$Obj_2 = \sum_{j=1}^{M} \frac{1}{2} \left(\sum_{i=2}^{N} SP[i] \cdot (X[i,j] - X[i-1,j]) + \sum_{i=1}^{N-1} SP[i+1] \cdot (Y[i+1,j] - Y[i,j])\right)$

Obj_3: Maximize the total number of facets mentioned within the extracted arguments.
$Obj_3 = \sum_{j=1}^{M} \left(\sum_{i=1}^{N} \left(\sum_{k=1}^{K} F[i,k]\right) \cdot (X[i,j] - Y[i,j])\right)$

Table 4: Objectives used in ArgExt-ILP

text segmentation techniques have been proposed like C99 algorithm (Choi, 2000) which identifies optimal segments, semantic segmentation technique (Alemi and Ginsparg, 2015) which incorporates semantic word embedding while identifying the segments. Some recent work using deep learning for text segmentation is by Lattisi et al. (2022) where they are using BERT model's Next Sentence Prediction (NSP) probability as a coherence score between sentences in their objective. Moro and Ragazzi (2022) employs self-segmentation technique to extract the semantic chunks from a long legal documents, where they fine-tuned the Legal-BERT model with metric learning setup to incorporate the segment semantics. Our technique is motivated by the work of Palshikar et al. (2019) which also uses the ILP framework for identifying certain types of sections in a document.

5 Experiments

5.1 Annotated Dataset for Evaluation

We identified 10 court judgements related to industrial disputes from the Supreme Court of India[5]. These judgements were annotated manually with gold-standard legal arguments[6] These 10 judgements contain 1524 sentences spread across 418 paragraphs overall. The total of 127 gold-standard arguments were identified during the manual annotation process. Each argument is represented by its start and end sentence numbers where each sentence in between is considered as a part of the argument. Annotators were also asked to identify a sentence for each argument which contains its major claim. To estimate the inter-annotator agreement (IAA), we used the pygamma-agreement library (Titeux and Riad, 2021) which is based on (Mathet et al., 2015). We used the positional dissimilarity based γ statistic for comparing the arguments identified by two annotators and the average value of γ was observed to be 0.405. As another estimate for IAA, we also used the same evaluation metrics (described in Section 5.3) which we use to evaluate the predicted arguments. The F1-scores for the IAA were observed as follows: Arg-exact=0.3, Arg-subset=0.47, Arg-overlap=0.56, and Arg-sentences=0.59. The IAA scores are not very strong which indicates the difficulty level and subjective nature of the task.

For training the classifiers needed for identifying the argument markers AS and SP, we used the ECHR corpus as it is similar to our dataset in the sense that it is also a corpus of court judgements which is annotated for legal arguments by lawyers. However, this corpus did not help in identifying claims with reasonable accuracy by training a classifier, hence we had to rely on the linguistic rules. This shows that even though this corpus is similar to our dataset, there are some differences, especially the language used for claim sentences.

5.2 Baselines

Baseline-TextSeg: We use C99 algorithm (Choi, 2000) for segmenting the court judgements. We retain only those text segments as legal arguments which contain at least one claim sentence, and discard all the remaining text segments.

Baseline-RhetoricalRoles: We obtained rhetorical roles for each sentence in each judgement using the opennyai python package[7] based on the work of Kalamkar et al. (2022). Each sequence of contiguous sentences which is labelled by the same argument indicating rhetorical role (ARG_RESPONDENT or ARG_PETITIONER) is identified as a legal argument.

Baseline-LegalArgs: This baseline is based on the technique proposed by Habernal et al. (2023) where a paragraph is given as an input to a sequence labelling model which labels each token in the paragraph with appropriate argument type using BIO encoding. For making it comparable with our problem setting, we merged all their 16 argument types into a single type, re-trained the *roberta-large* model on their training dataset, and used the model to infer the argument labels on each paragraph in our evaluation dataset. We also extended the token level classification output to sentence level, i.e., even if a subset of tokens in a sentence is labelled as part of an argument by the model, we consider the entire sentence as a part of the argument.

5.3 Evaluation Metrics

For evaluating the predicted arguments, we propose a set of new metrics. These are in the form of traditional precision, recall and F1-score scores only but they differ from each other in how true positives (TP), false positives (FP), and false negatives (FN) are computed based on when two arguments are

[5]Downloaded from http://www.liiofindia.org/in/cases/cen/INSC/

[6]The annotation guidelines are shared in Appendix C. The dataset would be shared upon request.

[7]https://pypi.org/project/opennyai/

Metric	Technique	With predicted claims			With gold-standard claims		
		P	R	F1	P	R	F1
Arg-exact	Baseline-LegalArgs (Habernal et al., 2023)	0.206	0.055	0.087	0.296	0.063	0.104
	Baseline-RhetoricalRoles Kalamkar et al. (2022)	0.012	0.016	0.014	0.031	0.016	0.021
	Baseline-TextSeg (Choi, 2000)	0.029	0.047	0.036	0.058	0.047	0.052
	ArgExt-Rules	0.088	0.094	0.090	0.257	0.142	0.183
	ArgExt-ILP	0.145	0.197	**0.167**	0.330	0.283	**0.305**
Arg-subset	Baseline-LegalArgs (Habernal et al., 2023)	0.417	0.118	0.184	0.576	0.150	0.238
	Baseline-RhetoricalRoles Kalamkar et al. (2022)	0.160	0.205	0.179	0.351	0.213	0.265
	Baseline-TextSeg (Choi, 2000)	0.251	0.433	0.318	0.434	0.441	0.438
	ArgExt-Rules	0.223	0.165	0.190	0.684	0.205	0.315
	ArgExt-ILP	0.380	0.551	**0.450**	0.641	0.661	**0.651**
Arg-overlap	Baseline-LegalArgs (Habernal et al., 2023)	0.500	0.134	0.211	0.667	0.142	0.234
	Baseline-RhetoricalRoles Kalamkar et al. (2022)	0.243	0.205	0.222	0.385	0.157	0.223
	Baseline-TextSeg (Choi, 2000)	0.251	0.409	0.311	0.447	0.362	0.400
	ArgExt-Rules	0.294	0.315	0.304	0.486	0.268	0.345
	ArgExt-ILP	0.427	0.575	**0.490**	0.690	0.598	**0.641**
Arg-sentences	Baseline-LegalArgs (Habernal et al., 2023)	0.470	0.129	0.203	0.739	0.140	0.235
	Baseline-RhetoricalRoles Kalamkar et al. (2022)	0.521	0.259	0.346	0.624	0.218	0.323
	Baseline-TextSeg (Choi, 2000)	0.403	0.708	0.514	0.529	0.616	0.569
	ArgExt-Rules	0.594	0.331	0.425	0.901	0.263	0.407
	ArgExt-ILP	0.506	0.768	**0.610**	0.758	0.752	**0.755**

Table 5: Evaluation results for argument extraction by various techniques

With predicted claims:				
Objective	**Arg-Exact**	**Arg-Subset**	**Arg-Overlap**	**Arg-Sentences**
$Obj_1 + Obj_2 - Obj_3$	0.167	0.450	0.490	0.610
Without Obj_1	0.106	0.352	0.397	0.525
Without Obj_2	0.168	0.427	0.474	0.606
Without Obj_3	0.173	0.448	0.502	0.612
With gold-standard claims:				
Objective	**Arg-exact**	**Arg-subset**	**Arg-overlap**	**Arg-sentences**
$Obj_1 + Obj_2 - Obj_3$	0.305	0.651	0.641	0.755
Without Obj_1	0.197	0.527	0.535	0.660
Without Obj_2	0.340	0.659	0.694	0.764
Without Obj_3	0.287	0.638	0.647	0.762

Table 6: Ablation study for objectives in ArgExt-ILP (F1-scores)

considered to be "matching" with each other. If a gold-standard argument "matches" with a predicted argument, then a TP is counted, otherwise a FN is counted. Further, if a predicted argument does not "match" with any gold-standard argument, then a FP is counted. The following metrics correspond to different ways of "matching":

Arg-exact: A predicted argument is considered to be "matching" with a gold-standard argument if their start and end sentence indices are same.

Arg-subset: A gold-standard argument is considered to be "matching" with a predicted argument if the set of sentence indices within the gold-standard argument is a proper subset of the set of sentence indices of the predicted argument.

Arg-overlap: Two arguments are considered to be "matching" with one another if Jaccard similarity between the sets of sentence indices within the two arguments is greater than or equal to 0.5.

Arg-sentences: Unlike the above metrics where TP/FP/FN are counted at argument-level, in this metric, these are counted at a sentence level. A sentence in any predicted argument is considered a TP if it is also part of some gold-standard argument, otherwise it is considered as a FP. Similarly, a sentence in a gold-standard argument is considered as a FN if it is not part of any predicted argument.

5.4 Evaluation Results

Table 5 shows the comparative performance of our proposed argument extraction techniques with respect to the baselines. It can be observed that ArgExt-ILP outperforms all other techniques across all evaluation metrics. Even though ArgExt-ILP and ArgExt-Rules are based on the same argument markers, ArgExt-ILP consistently outperforms ArgExt-Rules. This shows that the ILP framework is helpful in combining multiple weak evidences in the form of argument markers and potentially conflicting domain rules in a more princi-

With predicted claims:				
Constraints	**Arg-exact**	**Arg-subset**	**Arg-overlap**	**Arg-sentences**
All constraints in Table 3	0.167	0.450	0.490	0.610
Without C_6	0.080	0.418	0.416	0.530
Without C_7	0.147	0.423	0.450	0.601
Without C_8	0.157	0.459	0.472	0.604
Without C_9	0.060	0.548	0.244	0.540
With gold-standard claims:				
Constraints	**Arg-exact**	**Arg-subset**	**Arg-overlap**	**Arg-sentences**
All constraints in Table 3	0.305	0.651	0.641	0.755
Without C_6	0.086	0.435	0.422	0.535
Without C_7	0.352	0.641	0.656	0.746
Without C_8	0.347	0.679	0.694	0.772
Without C_9	0.132	0.649	0.395	0.598

Table 7: Ablation study of the constraints in ArgExt-ILP (F1 scores)

pled manner than a rule-based logic. However, the performance of ArgExt-ILP is still far from being perfect and this highlights the challenging nature of the task. The error analysis shows that there are mainly two sources of errors - (i) incorrect identification of claim sentences and (ii) incorrect boundary identification of the arguments. In order to estimate the effect of the first source, we re-run all the techniques assuming gold-standard claim sentences are known. Table 5 shows the detailed results in this setting in the last 3 columns. Again, ArgExt-ILP outperforms all other techniques and also improves significantly over its own performance with predicted claim sentences. This shows that there still scope for improvement in identification of argument markers like claims so as to improve the end-to-end argument extraction. More implementation details for ArgExt-ILP are provided in Appendix B.

Ablation Studies for ArgExt-ILP: Table 6 shows the results of ablation for the multiple objectives used in ArgExt-ILP. It can be observed that the objective Obj_1 is the most important one as the performance drops the most if we remove it. The objective Obj_2 is contributing when we are using predicted claim sentences which is a more practical setting, whereas the objective Obj_3 has mixed results across various metrics. Similarly, Table 7 shows the results of ablation studies for the multiple constraints used in ArgExt-ILP. The constraints C_6 and C_9 are the most significant ones as removing them results in reduced performance consistently.

Argument Markers Identification Accuracy: Table 8 shows the accuracy with which individual argument markers C, AS and SP are identified. It can be observed that individually these markers are not identified with very high accuracy and hence we are considering them as *weak evidences*.

Argument Marker	**P**	**R**	**F1**
C (linguistic rules)	0.422	0.724	0.533
AS ($prob \geq 0.5$)	0.356	0.612	0.450
SP ($prob \geq 0.2$)	0.577	0.653	0.613

Table 8: Evaluation results for argument markers

6 Conclusions and Future Work

We proposed a technique to extract legal arguments from Indian Supreme Court judgements by first identifying a set of certain *argument markers* and then incorporating them in an Integer Linear Programming (ILP) framework with domain knowledge based constraints. Individually, these argument markers are weak indicators of arguments mentioned in the text of a judgement, but the information from multiple such markers gets combined effectively in our ArgExt-ILP technique. We annotated a small dataset of 10 court judgements containing 127 legal arguments and evaluated our techniques on it along with multiple competent baselines. We demonstrated that ArgExt-ILP outperforms other baselines across multiple evaluation metrics. To the best of our knowledge, this is the first attempt to extract legal arguments from Indian court judgements and also a first arguments-annotated dataset for the same. As part of future work, our argument extraction techniques need to be improved further in multiple aspects – (i) the accuracy of identifying individual argument markers needs to be improved further which will automatically improve ArgExt-ILP's performance, (ii) we plan to do away with some of our simplifying assumptions to also extract overlapping and non-contiguous arguments, and (iii) we plan to evaluate our techniques on a wider variety of court judgements such as judgements other than industrial disputes and also from other geographies than India.

References

Alexander A Alemi and Paul Ginsparg. 2015. Text segmentation based on semantic word embeddings. *arXiv preprint arXiv:1503.05543*.

Basit Ali, Sachin Pawar, Girish Palshikar, and Rituraj Singh. 2022. Constructing a dataset of support and attack relations in legal arguments in court judgements using linguistic rules. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 491–500, Marseille, France. European Language Resources Association.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Freddy YY Choi. 2000. Advances in domain independent linear text segmentation. *arXiv preprint cs/0003083*.

Giulia Grundler, Piera Santin, Andrea Galassi, Federico Galli, Francesco Godano, Francesca Lagioia, Elena Palmieri, Federico Ruggeri, Giovanni Sartor, and Paolo Torroni. 2022. Detecting arguments in cjeu decisions on fiscal state aid. In *Proceedings of the 9th Workshop on Argument Mining*, pages 143–157.

Ivan Habernal, Daniel Faber, Nicola Recchia, Sebastian Bretthauer, Iryna Gurevych, Indra Spiecker genannt Döhmann, and Christoph Burchard. 2023. Mining legal arguments in court decisions. *Artificial Intelligence and Law*, pages 1–38.

Prathamesh Kalamkar, Aman Tiwari, Astha Agarwal, Saurabh Karn, Smita Gupta, Vivek Raghavan, and Ashutosh Modi. 2022. Corpus for automatic structuring of legal documents. In *Proceedings of the Language Resources and Evaluation Conference*, pages 4420–4429, Marseille, France. European Language Resources Association.

Tiziano Lattisi, Davide Farina, and Marco Ronchetti. 2022. Semantic segmentation of text using deep learning. *Computing and Informatics*, 41(1):78–97.

Yann Mathet, Antoine Widlöcher, and Jean-Philippe Métivier. 2015. The unified and holistic method gamma (γ) for inter-annotator agreement measure and alignment. *Computational Linguistics*, 41(3):437–479.

Raquel Mochales and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law*, 19:1–22.

Gianluca Moro and Luca Ragazzi. 2022. Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11085–11093.

Girish K Palshikar, Sachin Pawar, Rajiv Srivastava, and Mahek Shah. 2019. Identifying repeated sections within documents. *Computación y Sistemas*, 23(3):819–828.

Sachin Pawar, Basit Ali, Girish Palshikar, Ramandeep Singh, and Dhirendra Singh. 2023. Extraction and classification of statute facets using few-shots learning. In *19th International Conference on Artificial Intelligence and Law (ICAIL)*.

Prakash Poudyal. 2016. Automatic extraction and structure of arguments in legal documents. *Sarah A. Gaggl, Matthias Thimm (Eds.)*, page 19.

Prakash Poudyal, Jaromír Šavelka, Aagje Ieven, Marie Francine Moens, Teresa Goncalves, and Paulo Quaresma. 2020. Echr: Legal corpus for argument mining. In *Proceedings of the 7th Workshop on Argument Mining*, pages 67–75.

Piera Santin, Giulia Grundler, Andrea Galassi, Federico Galli, Francesca Lagioia, Elena Palmieri, Federico Ruggeri, Giovanni Sartor, Paolo Torroni, et al. 2023. Argumentation structure prediction in cjeu decisions on fiscal state aid. In *ICAIL'23: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages N–A.

Hadrien Titeux and Rachid Riad. 2021. pygamma-agreement: Gamma γ measure for inter/intra-annotator agreement in python. *Journal of Open Source Software*, 6(62):2989.

Dietrich Trautmann. 2020. Aspect-based argument mining. *arXiv preprint arXiv:2011.00633*.

Zhongyu Wei, Chen Li, and Yang Liu. 2017. A joint framework for argumentative text analysis incorporating domain knowledge. *arXiv preprint arXiv:1701.05343*.

Huihui Xu and Kevin Ashley. 2023. Argumentative segmentation enhancement for legal summarization. *arXiv preprint arXiv:2307.05081*.

Gechuan Zhang, Paul Nulty, and David Lillis. 2023. Argument mining with graph representation learning.

A Details about linguistic rules

Following are the complete details about various list of words used in by the linguistic rules for identification of claim sentences.

List of causal discourse markers used in LR1 and LR2: *therefore*, *thus*, *hence*, *consequently*, *moreover*, *furthermore*, *similarly*, *likewise*, *accordingly*, *thereby*

List of nouns used in LR3: *opinion*, *belief*, *impression*, *indication*, *judgement*, *assessment*, *estimation*, *position*, *argument*, *argumentation*, *submission*, *contention*, *objection*, *justification*, *conclusion*, *claim*, *clarification*.

List of verbs used in LR3: *sustain*, *contend*, *argue*, *debate*, *assert*, *conclude*, *assess*, *believe*, *maintain*, *submit*, *show*, *demonstrate*, *prove*, *appear*, *seem*, *clear*, *justify*, *conclude*, *claim*, *affirm*, *arrogate*, *indicate*, *clarify*, *hold*, *opine*. Also note that the list contains only the base forms of these verbs but while matching in the sentence, we consider all the morphological variations such as *conclude* ⇒ *concluded*, *concluding*, *concludes*.
List of negative adjectives used in LR4: *unfair*, *erroneous*, *incorrect*, *wrong*, *inaccurate*, *inexact*, *imprecise*, *invalid*, *fallacious*, *misleading*, *illogical*, *unsound*, *faulty*, *flawed*, *spurious*, *unfounded*, *unjustified*, *illegal*, *inappropriate*, *inconsistent*, *unsustainable*, *unwarranted*
List of positive adjectives used in LR4: *correct*, *accurate*, *exact*, *precise*, *valid*, *logical*, *justified*, *warranted*, *consistent*, *sustained*, *fair*, *legal*, *appropriate*, *permitted*, *maintainable*.
List of negative adverbs used in LR4: *inconsistently*, *unfairly*, *erroneously*, *incorrectly*, *wrongly*, *mistakenly*, *illegally*, *inappropriately*, *spuriously*.
List of positive adverbs used in LR4: *consistently*, *fairly*, *correctly*, *legally*, *appropriately*.

B Implementation Details

For solving ILP programs in ArgExt-ILP, we used the glpk solver[8] through Python's pyomo library[9]. For better running time efficiency, we split each judgement into two parts, solve two separate ILP programs, and later merge their solutions to get the final output. We used $M = 10$ so that at most 20 arguments can be extracted from each judgement. Also while splitting a judgement, we make sure that it is always split at a paragraph boundary. As there is a constraint (C_9) which ensures that no extracted argument can cross paragraph boundaries, we believe that this is a reasonable approximation.

C Annotation Guidelines

The following guidelines were shared with the annotators.
Goal: To identify legal arguments mentioned in court judgements. We assume each legal argument to be a chunk of contiguous sentences in the court judgement such that each chunk corresponds to a complete and coherent argument.
Annotation format: For each coherent and complete argument (consisting of a chunk of k contiguous sentences), the the following details are noted – **Filename** (file name of the court judgement), **StartSentNo** (sentence number of the first sentence of an argument), **EndSentNo** (sentence number of the last sentence of the argument), **ClaimSentNo** (sentence number of the sentence which contains the key claim/conclusion of the argument).

General guidelines:
1. Only contiguous sentences should be identified as an argument.
2. No overlapping arguments should be identified.
3. Each identified argument should be "complete" (as self-sufficient as possible to understand it) and "coherent" (should be mainly related to only one topic or legal point).
4. There should be at least one sentence in an argument which contains some "claim" being made or some "conclusion" being arrived at or some legal point be argued about. It also includes some opinion being expressed or some decision (or evaluation of lower court decision) that judge/court arrives at. Generally, the ultimate "claim" in an argument occurs either as the first sentence or the last sentence within the contiguous sentences identified as a legal argument. Some examples of "claims" are as follows:

- *the leniency shown by the Labour Court is clearly unwarranted and would in fact encourage indiscipline* (**evaluation of lower court decision**)
- *The finding is based on surmises* (**opinion**)
- *the petitioner who is working as an Area Sales Executive is not a workman within the meaning of Section 2(s) of the Industrial Disputes Act, 1947.* (**conclusion or legal point**)

Some examples of sentences which DO NOT contain any "claim":

- *A review application, however, was filed inter alia on the premise that the workmen were not entitled to claim any bonus.* (**a past event or fact**)
- *Section 12 of the Act provides the duties of the Conciliation Officer.* (**referring to a statute**)
- *This Court while allowing the appeal directed the respondent No.2 the Labour Commissioner, Chandigarh to make a reference under Section 12 of the Act.* (**direction by a court**)

Please note that the above are just some types of sentences which are not "claims" such as a past event, fact, direction by a court, or reference to a statutes, etc. There may be several additional types of sentences which are not "claims".
5. There should be at least one sentence in an argument which contains supporting facts, statements, evidences, or any other premises including prior

[8] https://www.gnu.org/software/glpk/
[9] https://pypi.org/project/Pyomo/

cases, statutes etc. which support the major "claim" or "conclusion" in the argument.
6. An argument may consist of a single sentence, i.e., both "claim" and its supporting premises are present in the single sentence.
7. Even if we are using the terminology "argument", the argument need not be made only by the contesting parties (appellant/plaintiff and respondent/defendant). The argument may correspond to reasoning given by lower court / current court to arrive at certain conclusion.
8. There can be multiple "claims" in an argument. But there exists only one major claim which may be supported by intermediate claims.
9. Opinion of any court (judge) can be considered as a claim. E.g., *the order of Labour Court as affirmed by High Court can not be sustained*
10. An argument can be found within sentences which are quoted from some prior case. That means the sentences are not about the current case but show why certain argument was made or decision was taken in a prior case.