Survey: Legal Argument Analysis

Jalay Shah, Anuj Srivastava and **Pushpak Bhattacharyya**
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
{jalay, anuj, pb}@cse.iitb.ac.in

Abstract

Legal Practitioners need to invest significant time in analyzing prior case Judgements (legal document containing the proceedings of a prior case). We can assist the legal practitioners in this task using Natural Language Processing. In this survey paper, we explain the problem statement in hand. We also discuss the work done for assisting the legal professionals in analyzing prior cases. The work discussed here span across different tasks framed by the researches to assist the legal practitioners.

1 Problem Statement

A lot of decisions are made in the Courts based on analyzing relevant prior cases and finding out the similarities between the prior case and current case as well as reasoning whether the decision of the prior case should be applied in the current case or not. As a result, legal practitioners require to invest a lot of time and effort in analyzing prior relevant cases. Main objective is to develop a system to semi-automate the process of analyzing a given prior case.

2 Motivation

NLP is a vast field, containing many tools and techniques for language processing. Many of such tools have been applied to text of varying domains and proportions. One such area is of legal domain a.k.a. LegalNLP.

LegalNLP is important because it can help lawyers save time and effort by reducing the amount of work they have to do. Many legal tasks necessitate the experience of legal professionals as well as a thorough comprehension of numerous legal papers. Even for legal professionals, retrieving and comprehending legal documentation takes a long time. A system trained in LegalNLP can reduce the time spent on these time-consuming tasks and help the legal professionals. Furthermore, LegalNLP can also assist, in understanding legal documents, for those who are unfamiliar with the legal field.

Prior cases and statutes are two main pillars of the Indian Judicial System. A lot of decisions are made in the Courts by analyzing relevant prior cases and statutes. Legal practitioners are required to find out the similarities between any prior case and the current case as well as reasoning on whether the decision of the prior case is applicable in the current case or not. If it is applicable, then on what basis/argument was the decision taken in the prior case. Finally applying the same argument in the current case. Each prior case has a Judgement. A Judgement is a document that is written by the Court containing the arguments and counter-arguments put forth during the proceeding in the Court. It is important to understand the content of the Judgements in order to understand a prior case. As a result, legal practitioners are required to invest a lot of time and effort in analyzing prior relevant cases by reading their judgements. This time spent in processing the information pertaining to a case adds to the pendency in the legal system.

On the other hand, a large pendency is present in the legal Indian Judiciary system. One of the main challenges facing the Judicial System of India is the pendency of cases. As of 2021, there are about 73,000 cases pending before the Supreme Court and about 44 million in all the courts of India. It is of utmost importance to solve this problem of reducing the pendency as a great number of people benefit from the same. The field of LegalNLP deals with developing techniques to automate a wide variety of tasks. Hence, it becomes a very good candidate which has the potential to achieve our task of pendency reduction.

3 Literature Survey

This section will give a literature survey of the tasks and techniques which are developed to assist legal professionals in understanding the argument flow of a prior case.

3.1 Prior Case Retrieval

Prior case retrieval is a very important task in the legal domain. The legal practitioners/lawyers often have to search through prior cases to find out relevant cases. This takes a lot of effort and time. Prior case retrieval strives to automate this task. This task can be framed in multiple ways. One such way is as follows. The user only needs to give a natural language query. The system then processes this query and outputs a set of cases that are relevant for the said query. Some of the examples of such a query (Ghosh et al., 2020) are as follows :

> *What are the cases where blood stains were found on clothes of the deceased?*
>
> *What are the cases where the police murdered the deceased?*
>
> *What are the cases where a husband has set his wife on fire?*
>
> *Which are the cases where the appellant demanded money?*
>
> *Which are the cases where the appellant assaulted the deceased?*
>
> *Which are the cases where the respondent has forged signatures?*

We will now explain the 2 relevant techniques which tackle the above task.

3.1.1 Using Witness Testimonies

Witness testimonies are a type of evidence that is obtained from a witness who makes a solemn statement or declaration of fact. For eg, one testimony can be :

> *The body of Gian Kaur was sent to Dr. Singh (PW 6) for post-mortem who noticed five minor injuries on the body of the deceased.*

Such testimonies play a very important role in deciding the final decision. The importance of such testimonies (as stated in Ghosh et al. (2020)) is given in the below paragraph.

Witness testimonies and their cross-examinations by the counsels have a significant effect on the judges' decision. Judges often comment in the judgment on (a) the correctness, quality, completeness, and reliability of the testimonies of a witness; (b) the interrelationships between the testimonies of various witnesses (e.g., consistency or contradictions); and (c) the impact ("weighing in") of various witness testimonies on their final decision. The specific contents of witness testimonies and such high-level analyses are valuable for preparing a case, retrieving relevant past cases, understanding the strengths and weaknesses of a case, predicting court decisions, and extracting legal argumentation.

Such testimonies are also present in abundance in court judgments. Hence we can infer that if a system is developed to automatically extract such witness testimonies, then it can be used for prior case retrieval. This is the main goal of Ghosh et al. (2020). They have developed 2 systems, one rule-based and another based on deep learning models, to extract witness testimonies and retrieve prior cases using them. They are described below.

Upon further exploration of Court judgements, it can be found that the testimonies often obey a structure. Ghosh et al. (2020) exploited this structure to develop a rule based algorithm to extract all such sentences which contained witness testimonies. The rules which were used are given below for reference :

1. Presence of explicit (e.g., eye-witness, P.W.2) or implicit witness mentions.
2. Implicit mentions can be pronouns (*he*, *she*), person-indicating common nouns (*landlord*, *doctor*), or actual person names (*S.I. Patil*).
3. Presence of at least one statement-indicating verb like stated, *testified*, *narrated.*
4. Within its dependency sub-tree, the statement verb should contain at least one

of the following: a clausal complement (*ccomp*) or open clausal complement (*xcomp*).

5. The statement verb should NOT have a child which negates it like not.

6. The statement verb should have at least one witness mention within its *nsubj* or agent dependency subtree (to ensure that the witness mention is subject/agent of the statement verb) but should NOT have any legal role (e.g. lawyer, counsel, judge) mention within its *nsubj* or agent dependency sub-tree (to exclude the statements by lawyers or judges).

One may notice that the rules are very specific and hence brittle. They won't be able to detect testimonies that differ a bit in their sentence structure. For eg.

> *PW-15 further deposed that she knew Bharosa Colour Lab as she had been there several times to meet Mahesh*

The testimony given above cannot be detected using above mentioned rules. Hence to make the system more robust, they used an LSTM based classifier trained on the sentences extracted using rule-based method. The LSTM takes one word as input in one timestep and outputs one label in the final timestep to classify the sentence as testimony/not-a-testimony.

After extracting witness testimonies, we also need to use it to retrieve prior cases given a query. This problem was approached using Semantic Role Labelling (SRL). SRL includes finding the main verb (predicate) of the sentence as well as finding all the arguments of this main verb. Many arguments may or may not be present, so the focus was on the main 2 arguments i.e. A0 (subject of the verb) and A1 (object of the verb).

To match a query with its relevant case, all possible semantic roles were extracted from the case and similarity search was done for the tuple (Predicate, A0, A1) as given below :

$$SIM(Q, D) = max_S(cos\_sim(Repr(Q), Repr(S))$$

where Q is the query, D is the prior case and S is the tuple. $Repr()$ is a denoising autoencoder that provides a semantic representation of the tuples as a N-dim vector. This autoencoder was trained using the set of all the extracted tuples from all the judgments as the training set.

The results are shown in Table 1 given below.

We can see that this approach is performing significantly better than the baselines. Moreover, another benefit of the method is that it is highly interpretable compared to other methods. For eg., the system will give a high score only for those cases whose frame representation is very near to the frame representation of the query. For eg., the system gives a high score for the case containing the sentence

> *P.W. 1 to 5 have stated that the appellant **assaulted** the deceased with a crowbar on his head.*

when the query was

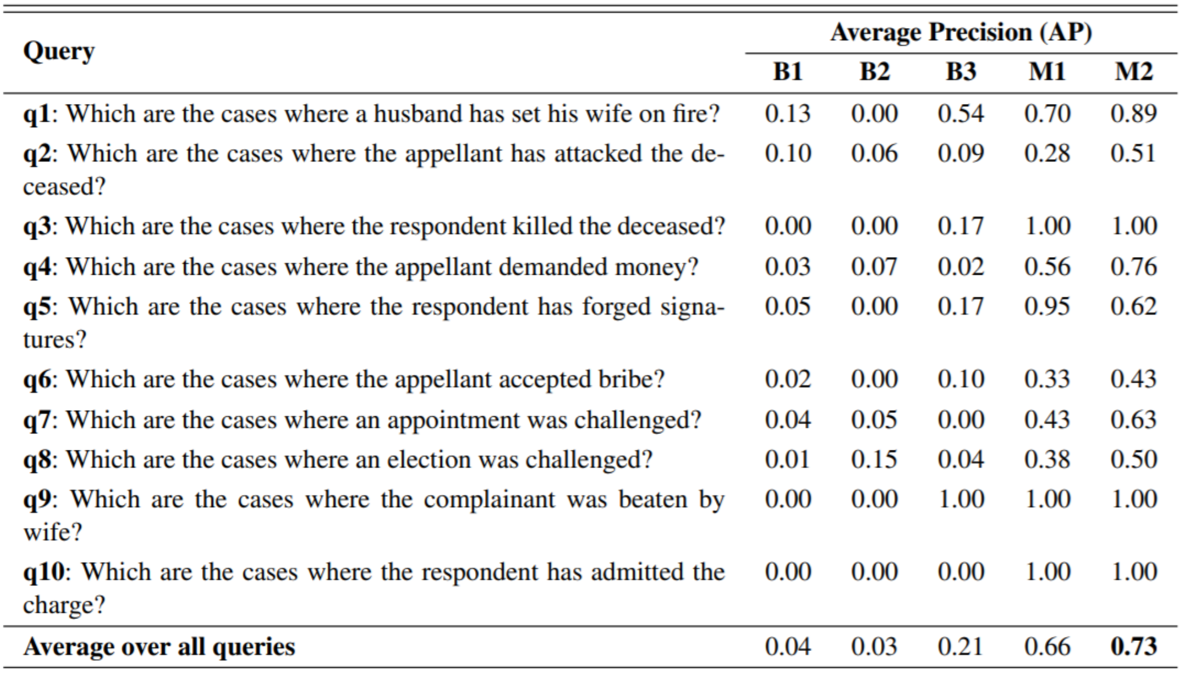
> *Which are the cases where the appellant has **attacked** the deceased?*

which shows that the system gives large importance to word similarity, thereby resulting in high interpretability.

3.1.2 Using Evidence

Just like witness testimonies, evidence is also an important part of the case. They have a large influence on the final decision. Moreover, they are a superset of witness testimonies, thereby covering a larger type of queries compared to the latter.

Ali et al. (2021) expands the idea of using witness testimonies to all types of evidence present in the judgments. A special structural representation may be required to extract evidence as it is much more diverse compared to testimonies. This is accomplished using Semantic Role Labelling (SRL). The overall structure is named as the evidence information model. It is divided into 2 parts i.e. Evidence Frame and the Observational Frame. Observational Frame contains information related to the observer, who observed a certain action taking place. This observer is no one but the witness, who gave testimony in the court. The Observational Frame mainly consists of 3 parts, given below :



Query	Average Precision (AP)				
	B1	**B2**	**B3**	**M1**	**M2**
q1: Which are the cases where a husband has set his wife on fire?	0.13	0.00	0.54	0.70	0.89
q2: Which are the cases where the appellant has attacked the deceased?	0.10	0.06	0.09	0.28	0.51
q3: Which are the cases where the respondent killed the deceased?	0.00	0.00	0.17	1.00	1.00
q4: Which are the cases where the appellant demanded money?	0.03	0.07	0.02	0.56	0.76
q5: Which are the cases where the respondent has forged signatures?	0.05	0.00	0.17	0.95	0.62
q6: Which are the cases where the appellant accepted bribe?	0.02	0.00	0.10	0.33	0.43
q7: Which are the cases where an appointment was challenged?	0.04	0.05	0.00	0.43	0.63
q8: Which are the cases where an election was challenged?	0.01	0.15	0.04	0.38	0.50
q9: Which are the cases where the complainant was beaten by wife?	0.00	0.00	1.00	1.00	1.00
q10: Which are the cases where the respondent has admitted the charge?	0.00	0.00	0.00	1.00	1.00
Average over all queries	0.04	0.03	0.21	0.66	**0.73**

Table 1: Average Precision with baseline models

- **ObserverVerb or OV :** The verb indicating the observation/discovery/disclosure (e.g., *found, revealed, stated*)

- **ObserverAgent or A0 :** The source disclosing the information (e.g., *person, agency, authority*)

- **EvidenceObject or EO :** The Evidence Object in focus (e.g., *post-mortem, report, FIR, letter*)

The Evidence Frame is the main part of the model, which contains all the information of the evidence in form of arguments of the main verb i.e. EvidenceVerb. The structure of the Evidence Frame is also provided below for reference :

- **EvidenceVerb or EV :**
  The main verb of any action, event or fact mentioned in a sentence or revealed by the Evidence Object *(e.g., killed, forged, escaped)*

- **Agent or A0 :**
  Someone who initiates the action indicated by the EvidenceVerb *(e.g., the accused, Ram, ABC Pvt. Ltd., etc.)*

- **Patient or A1 :**
  Someone who undergoes the action indicated by the EvidenceVerb. *(e.g., the deceased, a cheque of Rs. 3,200, his wife)*

- **Location or LOC :**
  Location where the action took place *(e.g., in the bedroom, at the bank, in Malaysia)*

- **Time or TMP :**
  Timestamp of the action *(e.g., about 12 hours back, in the morning, on Monday)*

- **Cause or CAU :**
  Cause of the action *(e.g., due to dowry, as a result of the CBI inquiry, out of sheer spite)*

- **Manner or MNR :**
  The manner in which the action took place *(e.g., as per the challan, fraudulently, wilfully)*

Any sentence S should satisfy the following conditions to be identified as an Evidence Sentence:

1. S should contain at least one Evidence Object. The list of words corresponding

to evidence objects is created automatically by using WordNet hypernym structure. It contains all words for which the following WordNet synsets are ancestors in hypernym tree – *artifact (e.g., gun, clothes), document (e.g., report, letter), substance (e.g., kerosene, blood)*

2. S should contain at least one action verb from a pre-defined set of verbs like *tamper, kill, sustain, forge* OR S should contain at least one observation verb from a pre-defined set of verbs like *report, show, find*. Both the pre-defined sets of verbs are prepared by observing multiple example sentences containing evidence objects.

3. In the dependency tree of S the evidence object (identified by E-R1) should occur within the subtree rooted at the action or observation verb (identified by E-R2) AND there should not be any other verb (except auxiliary verbs like *has been, was, were. is*) occurring between the two. This ensures that the evidence object always lies within the verb phrase headed by the action or observation verb.

The rules are very brittle, resulting in high precision, low recall dataset. To increase recall, LSTM based classifier is trained to determine the presence of evidence as well as testimony in a given sentence, resulting in a multi-label sentence classification task.

This increases recall, which can be proved by observing the list given below, which contains all such sentences which were detected by LSTM but not by rules.

1. *Raju PW2 took Preeti into the bathroom at the instance of Accused No. 1 who cut a length of wire of washing machine and used it to choke her to death, wh0 however, survived.*

2. *Raju PW2 took Satyabhamabai in the kitchen where the accused No. 1 had already reached and was washing the blood-stained knife.*

3. *Hemlata was also killed by inflicting knife injuries.*

4. *Accused No. 2 and Raju PW2 took the child into the room where Meerabai was lying dead in the pool of blood.*

5. *Blood-stained clothes of Accused No. 2 were put in the air-bag along with stolen articles.*

For prior case retrieval, a linear combination of 2 components is taken. One of them is the sentence-BERT (Reimers and Gurevych, 2019) which finds the cosine similarity between query and evidence containing sentence. Another is the semantic matching algorithm, which calculates the similarity by multiplying the similarity scores obtained by calculating the cosine similarity of phrase vectors of all the arguments. These phrase vectors are obtained by a linear combination of word embeddings of all the words which are present in the phrase. The process is explained clearly in Table 2 given below.

3.1.3 Results/Analysis

The results are shown in Table 3 given below.

The list of queries used for evaluation are as follows :

Q_1 : *blood stains were found on the clothes of the deceased.*

Q_2 : *the deceased had attacked some person with sticks.*

Q_3 : *the police has murdered the deceased.*

Q_4 : *some evidence shows that exhibited gun was not used.*

Q_5 : *the autopsy report reveals that some poisonous compunds were found in the stomach of the deceased.*

Q_6 : *the deceased is attacked with a knife.*

Q_7 : *a letter by the deceased reveal that dowry was demanded.*

Q_8 : *a cheque was dishonoured due to insufficient funds.*

Q_9 : *bribe was demanded by the police.*

Q_{10} : *signature was forged on the affidavit.*

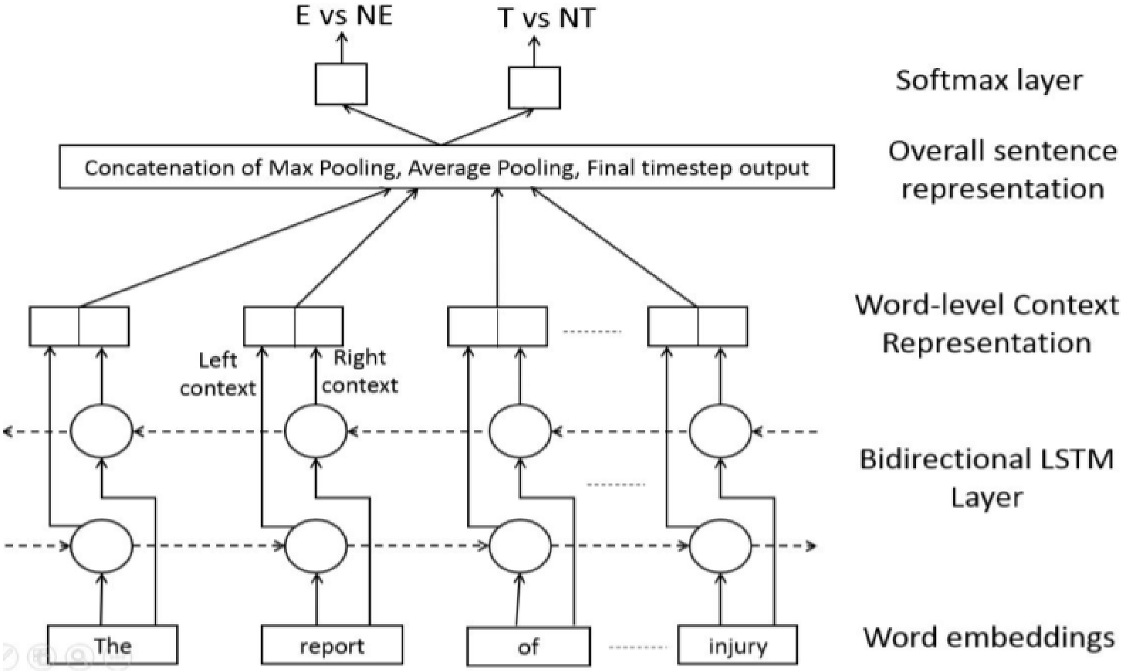

E vs NE
T vs NT
Softmax layer
Concatenation of Max Pooling, Average Pooling, Final timestep output
Overall sentence representation
Word-level Context Representation
Left context
Right context
Bidirectional LSTM Layer
The
report
of
injury
Word embeddings

Figure 1: Architecture of multi-label Bi-LSTM classifier

Analysis was also done for all the erroneous cases. Main 3 reasons were identified in the analysis, which are as follows :

- Some of the arguments were not present in the sentences.
- Some of the arguments contained unresolved co-references. For eg. take the sentences

  > *Instead of surrendering before the police, the deceased had attempted to kill the police. In retaliation, he was shot by them.*

  Here *them* and *he* are unresolved co-references, thereby resulting in low similarity scores.
- Low accuracy of cosine similarity acting as a bottleneck. For eg. *some poisonous compounds* gives a higher similarity score when matched with *three pieces of pellet* compared to *a heavy concentration of arsenic.*

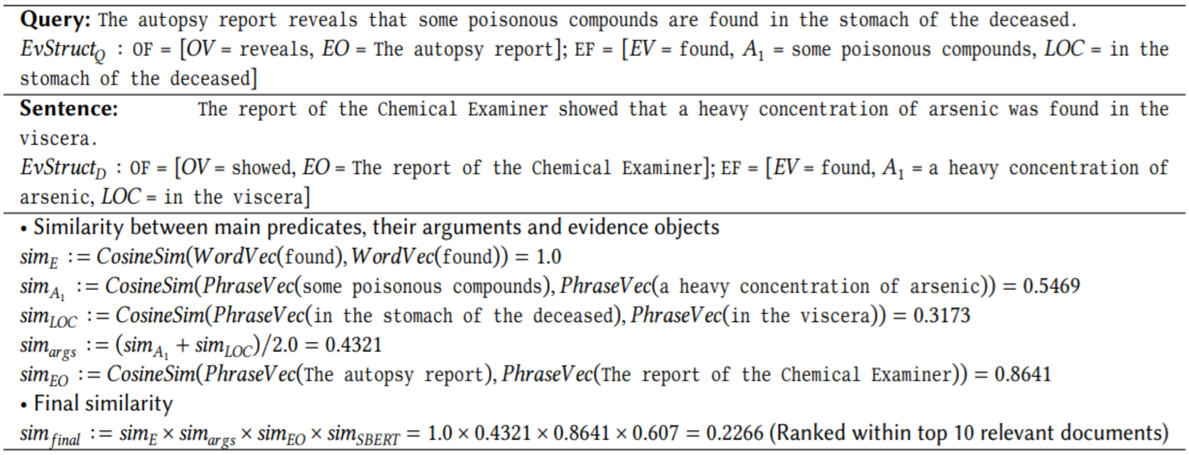
3.2 Rhetorical Role Classification

The task of rhetorical role classification is a sentence classification task, which segments the text present in the whole judgment into 7 rhetorical roles as explained in the previous chapter.

One of the techniques proposed for tackling this task is explained below.

Saravanan et al. (2008) proposes to frame this task as sequence labeling on sentences. Here the context for a sentence is all the previous sentences present in the judgment. They train a CRF model to classify the sentences into their corresponding rhetorical roles. The final output is obtained using the Viterbi algorithm to find the best path over all possible paths/sequences. The list of features used for training the CRF are given below :

- **Indicator/cue phrases**
  The term 'cue phrase' indicates the key phrases frequently used which are the indicators of common rhetorical roles of the sentences (e.g. phrases such as *We agree with court*, *Question for consideration is*, etc.,).
- **Named entity recognition**
  Named entities that are frequently mentioned in legal text like *Supreme Court, Lower court* etc., are taken into consider-



Query: The autopsy report reveals that some poisonous compounds are found in the stomach of the deceased.
$EvStruct_Q$: OF = [OV = reveals, EO = The autopsy report]; EF = [EV = found, A_1 = some poisonous compounds, LOC = in the stomach of the deceased]

Sentence: The report of the Chemical Examiner showed that a heavy concentration of arsenic was found in the viscera.
$EvStruct_D$: OF = [OV = showed, EO = The report of the Chemical Examiner]; EF = [EV = found, A_1 = a heavy concentration of arsenic, LOC = in the viscera]

• Similarity between main predicates, their arguments and evidence objects
$sim_E := CosineSim(WordVec(\text{found}), WordVec(\text{found})) = 1.0$
$sim_{A_1} := CosineSim(PhraseVec(\text{some poisonous compounds}), PhraseVec(\text{a heavy concentration of arsenic})) = 0.5469$
$sim_{LOC} := CosineSim(PhraseVec(\text{in the stomach of the deceased}), PhraseVec(\text{in the viscera})) = 0.3173$
$sim_{args} := (sim_{A_1} + sim_{LOC})/2.0 = 0.4321$
$sim_{EO} := CosineSim(PhraseVec(\text{The autopsy report}), PhraseVec(\text{The report of the Chemical Examiner})) = 0.8641$
• Final similarity
$sim_{final} := sim_E \times sim_{args} \times sim_{EO} \times sim_{SBERT} = 1.0 \times 0.4321 \times 0.8641 \times 0.607 = 0.2266$ (Ranked within top 10 relevant documents)

Table 2: Semantic Matching Algorithm

Query	$BM25_{\text{all}}$	$BM25_T$	$BM25_E$	$BM25_{TE}$	SB_T	SB_E	SB_{TE}	SM_T	SM_E	SM_{TE}
Q_1	0.24; 0.26	0.06; 0.02	$\underline{0.59}$; 0.49	$\underline{0.59}$; $\underline{0.52}$	0.00; 0.01	0.24; 0.15	0.18; 0.14	0.00; 0.01	0.24; 0.16	0.24; 0.14
Q_2	0.25; $\underline{0.43}$	0.00; 0.05	0.00; 0.04	0.00; 0.06	0.00; 0.01	0.00; 0.00	0.00; 0.00	0.25; 0.14	0.25; 0.25	$\underline{0.50}$; 0.30
Q_3	0.00; 0.01	0.00; 0.03	$\underline{0.33}$; 0.33	0.33; 0.35	$\underline{0.33}$; 0.12	0.00; 0.00	0.00; 0.09	$\underline{0.33}$; 0.12	0.00; 0.00	$\underline{0.33}$; 0.12
Q_4	0.17; 0.06	0.00; 0.01	0.00; 0.02	0.00; 0.04	0.00; 0.01	0.42; 0.25	0.42; 0.22	0.08; 0.04	0.25; 0.27	0.33; 0.29
Q_5	0.30; 0.43	0.10; 0.05	0.40; 0.35	0.40; 0.37	0.20; 0.15	$\underline{0.70}$; 0.80	$\underline{0.70}$; 0.80	0.00; 0.02	0.40; 0.40	0.40; 0.40
Q_6	0.31; 0.42	0.33; 0.28	0.38; 0.35	$\underline{0.46}$; 0.52	0.23; 0.14	0.33; 0.38	0.36; 0.40	0.20; 0.18	0.28; 0.27	0.41; 0.42
Q_7	0.25; 0.35	0.00; 0.08	$\underline{0.50}$; 0.54	0.50; 0.33	0.00; 0.04	0.00; 0.12	0.00; 0.09	0.25; 0.06	0.00; 0.00	0.25; 0.06
Q_8	0.48; 0.46	0.01; 0.09	0.67; 0.71	$\underline{0.71}$; $\underline{0.73}$	0.05; 0.02	0.62; 0.67	0.62; 0.67	0.00; 0.00	0.57; 0.63	0.57; 0.64
Q_9	0.20; 0.23	0.20; 0.17	0.20; 0.21	0.40; 0.31	0.40; 0.39	0.20; 0.21	$\underline{0.50}$; $\underline{0.51}$	0.40; 0.41	0.10; 0.12	$\underline{0.50}$; 0.48
Q_{10}	$\underline{0.50}$; 0.52	0.00; 0.11	0.25; 0.16	0.25; 0.21	0.00; 0.01	0.00; 0.04	0.00; 0.03	0.25; 0.13	$\underline{0.50}$; $\underline{0.61}$	$\underline{0.50}$; $\underline{0.61}$
Av	0.27; 0.32	0.08; 0.09	0.33; 0.32	0.36; 0.34	0.12; 0.09	0.25; 0.26	0.28; 0.30	0.18; 0.11	0.26; 0.27	$\underline{0.40}$; $\underline{0.35}$

Table 3: Comparison with baselines

ation and binary-valued entity type features are generated for the same.

- **Local features and Layout features** Includes arbitrary features like the presence of abbreviations, layout features such as the position of paragraph beginning, as well as the sentences appearing with quotes, etc.

- **State Transition features** Includes state transition features corresponding to the appearance of years attached with Section and Act nos. related to the labels arguing the case and arguments, Legal vocabulary features (obtained from basic vocabularies from training data), presence of capitalizations, affixes, etc. Also includes phrases that include *v.* and *act/section* which are the salient features for arguing the case and arguments categories.

The accuracy scores are provided in Table 4.

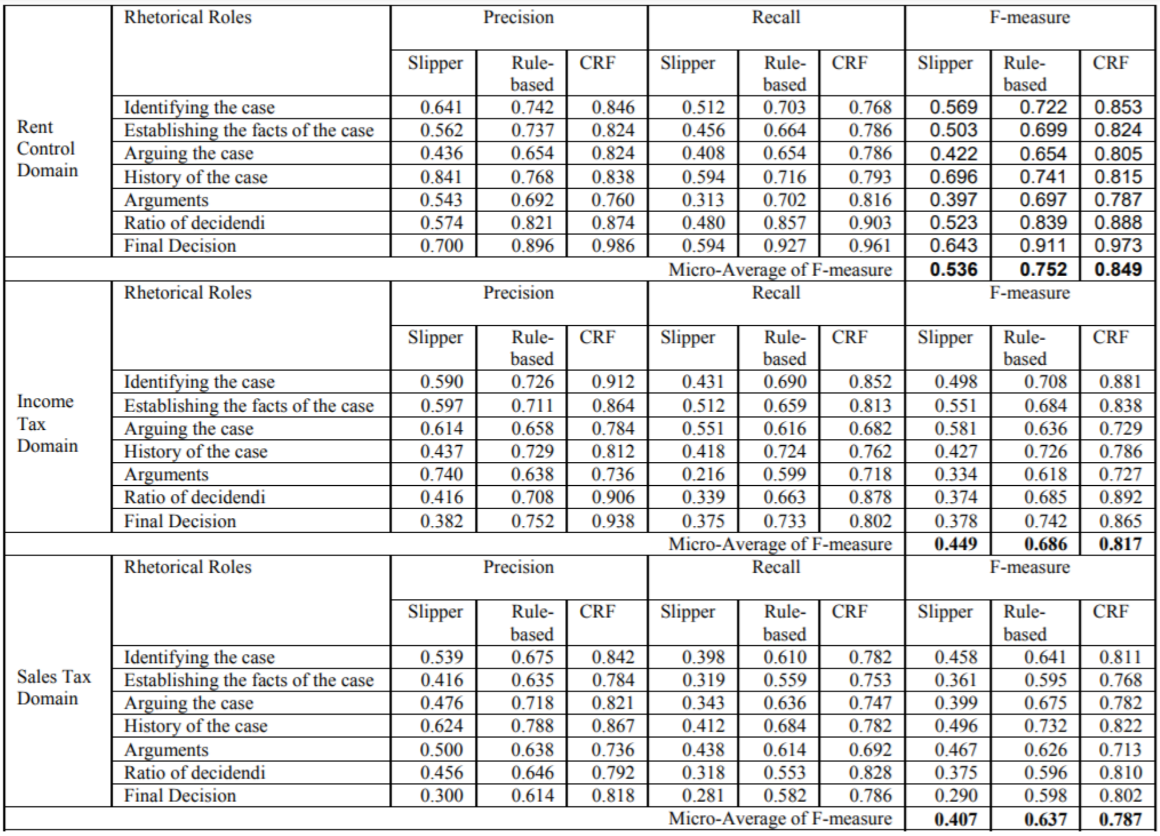
We see that compared to baseline methods, the CRF is performing significantly better, indicating the importance of incorporating context into role prediction.

Improvements were made in the above work using the deep learning models (Bhattacharya et al., 2019). The CRF was used on top of a Hierarchical Bi-LSTM model which resulted in an overall increase in the accuracy. The accuracy metrics are given in Table 5.

Observe that the model performs the best in predicting the Ratio and Ruling by Present Court (RPC), Ratio being the most frequent among all rhetorical roles and Ruling by the Present Court always present in a fixed position at the end of judgment. The model performs satisfactorily for all other labels, except 'Arguments', as they are interleaved with other labels.

3.3 Legal Argumentation Mining

This section will introduce and explain the field of argumentation mining from point of



	Rhetorical Roles	Precision			Recall			F-measure		
		Slipper	Rule-based	CRF	Slipper	Rule-based	CRF	Slipper	Rule-based	CRF
Rent Control Domain	Identifying the case	0.641	0.742	0.846	0.512	0.703	0.768	0.569	0.722	0.853
	Establishing the facts of the case	0.562	0.737	0.824	0.456	0.664	0.786	0.503	0.699	0.824
	Arguing the case	0.436	0.654	0.824	0.408	0.654	0.786	0.422	0.654	0.805
	History of the case	0.841	0.768	0.838	0.594	0.716	0.793	0.696	0.741	0.815
	Arguments	0.543	0.692	0.760	0.313	0.702	0.816	0.397	0.697	0.787
	Ratio of decidendi	0.574	0.821	0.874	0.480	0.857	0.903	0.523	0.839	0.888
	Final Decision	0.700	0.896	0.986	0.594	0.927	0.961	0.643	0.911	0.973
						Micro-Average of F-measure		**0.536**	**0.752**	**0.849**
	Rhetorical Roles	Precision			Recall			F-measure		
		Slipper	Rule-based	CRF	Slipper	Rule-based	CRF	Slipper	Rule-based	CRF
Income Tax Domain	Identifying the case	0.590	0.726	0.912	0.431	0.690	0.852	0.498	0.708	0.881
	Establishing the facts of the case	0.597	0.711	0.864	0.512	0.659	0.813	0.551	0.684	0.838
	Arguing the case	0.614	0.658	0.784	0.551	0.616	0.682	0.581	0.636	0.729
	History of the case	0.437	0.729	0.812	0.418	0.724	0.762	0.427	0.726	0.786
	Arguments	0.740	0.638	0.736	0.216	0.599	0.718	0.334	0.618	0.727
	Ratio of decidendi	0.416	0.708	0.906	0.339	0.663	0.878	0.374	0.685	0.892
	Final Decision	0.382	0.752	0.938	0.375	0.733	0.802	0.378	0.742	0.865
						Micro-Average of F-measure		**0.449**	**0.686**	**0.817**
	Rhetorical Roles	Precision			Recall			F-measure		
		Slipper	Rule-based	CRF	Slipper	Rule-based	CRF	Slipper	Rule-based	CRF
Sales Tax Domain	Identifying the case	0.539	0.675	0.842	0.398	0.610	0.782	0.458	0.641	0.811
	Establishing the facts of the case	0.416	0.635	0.784	0.319	0.559	0.753	0.361	0.595	0.768
	Arguing the case	0.476	0.718	0.821	0.343	0.636	0.747	0.399	0.675	0.782
	History of the case	0.624	0.788	0.867	0.412	0.684	0.782	0.496	0.732	0.822
	Arguments	0.500	0.638	0.736	0.438	0.614	0.692	0.467	0.626	0.713
	Ratio of decidendi	0.456	0.646	0.792	0.318	0.553	0.828	0.375	0.596	0.810
	Final Decision	0.300	0.614	0.818	0.281	0.582	0.786	0.290	0.598	0.802
						Micro-Average of F-measure		**0.407**	**0.637**	**0.787**

Table 4: Accuracy and F1 Scores for all the 3 domains

	FAC	ARG	Ratio	STA	PRE	RPC	RLC	Macro Average (across categories)
Constitutional	0.903	0.659	0.909	0.832	0.904	0.857	0.85	0.845
Labour & Industrial Law	0.776	0.505	0.929	0.423	0.728	0.783	0.681	0.689
Criminal	0.836	0.567	0.945	0.689	0.891	0.917	0.865	0.816
Land & Property	0.847	0.624	0.908	0.841	0.845	0.98	0.778	0.832
Intellectual Property	0.832	0.607	0.927	0.824	0.901	0.964	0.886	0.849
Macro Average (across labels)	0.8388	0.5924	0.9236	0.7218	0.8538	0.9002	0.812	–

Table 5: Accuracy and F1-Scores for all the rhetorical roles

view of legal text. It will also introduce the relevant techniques which were proposed by the researchers to model this task. Palau and Moens (2009) is the first paper to introduce argumentation mining as a field of legal NLP. In their paper, they also explain what argument is and establish the importance of argumentation mining.

Argumentation is the process of creating arguments that interact with other arguments, possibly of the opposite claim but on the same topic. A simple argument can be defined as the collection of premises and claims. Premises are the pieces of evidence that support the argument. Premises are known to be true by all the parties involved in the argument. All the premises together support the claim made by the author of that argument, also called a conclusion. A complex argument may contain other complex or simple arguments as premises. There is always a single conclusion per single argument. Thus argument can be defined as a conclusion, which is supported by a set of premises and sub-arguments.

Argumentation plays an important role in many areas. Many professionals, e.g. scientists, lawyers, journalists, or managers, routinely undertake argumentation as an integral part of their work, where to make an optimized decision in a certain situation, they need to analyze the pros and cons of a certain potential action which they believe leads to an optimal decision and further present it to the other parties involved in taking the decision to convince them of the optimality of the action. Furthermore, the study of argumentation is crucial in many NLP tasks. For example, reasoning agents need to communicate with each other and apply argumentation-based reasoning mechanisms to resolve the conflicts arising from their different views of goals, beliefs, and actions. Therefore, it becomes crucial to understand argumentation.

3.3.1 Types of arguments

The structure of arguments may vary depending on the style of parties involved in argumentation along with the situation wherein the argumentation is happening. Argumentation theory is a linguistic field which solely deals with the types of structures identified in man-made arguments.

The simplest theory of argumentation (Andone, 2005) divides the arguments in to 3 types, given below :

- **Simple Argumentation** The simple argumentation consists of a pair of 2 elementary units, one is the claim or conclusion and other the premise supporting that claim. Example given below.

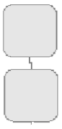
  

  Figure 2: Simple Argumentation

  > *The sky is cloudy. It will be rain soon.*

- **Multiple Argumentation** Multiple Argumentation consists of a claim and multiple standpoints (can be a premise or a sub-argument) supporting the claim independently are given in the argument. An example is given below.

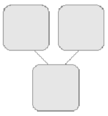
  

  Figure 3: Multiple Argumentation

  > *Postal deliveries in Holland are not perfect. You cannot be sure that a letter will be delivered the next day, that will be delivered to the right address, or that it will be delivered early in the morning.*

- **Compound Argumentation** This type of arguments consists of chain of (sub)arguments which reinforce each other. They are divided into 2 parts :

  - **Subordinatively Compound Argumentation** Here the arguments are connected in a chain, wherein each argument reinforces the argument present next in the chain in a linear fashion. Example given below.

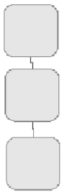
    

    Figure 4: Subordinatively Compound Argumentation

    > *She won't worry about the exam. She's bound to pass. She's never failed.*

  - **Coordinatively Compound Argumentation** Here the arguments are connected in parallel, wherein each argument partially reinforces the original claim. As a result, all the arguments when combined together support the claim, which is in direct contrast to the Multiple Argumentation scheme discussed above. An example is given below.

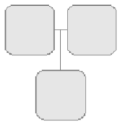
    

    Figure 5: Coordinatively Compound Argumentation

    > *This book has literary qualities: the plot is original, the story is*

> *well-told, the dialogues are incredibly natural, and the style is superb.*

The authors of Palau and Moens (2009) and Poudyal et al. (2020) focus only on the Multiple Argumentation scheme for the legal domain to reduce the complexity of the task.

3.3.2 Techniques

There are mainly 2 techniques developed for legal argumentation mining. They will be discussed below. There are 2 datasets for legal argumentation mining, out of which Poudyal et al. (2020) is public. It treats clauses as the elementary units of argumentation. It contains clauses of judgments segmented into either argumentative or non-argumentative. The argumentative units then form an intricate tree-like structure, where the root of the tree becomes a conclusion, and the child nodes of the root can be sub-arguments (if sub-tree is present) or premises (if leaf node is present). The sub-arguments always support the parent claim, unlike some of the work on non-legal domains where the premise either supports or attacks the claim. The argumentation tree structure is shown in Figure 6. The task of argumentation mining can be subdivided into 3 tasks :

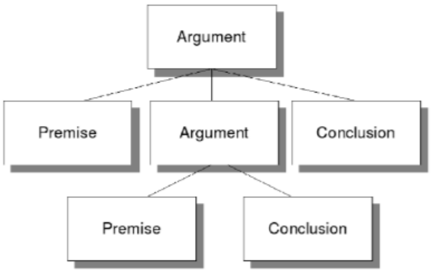


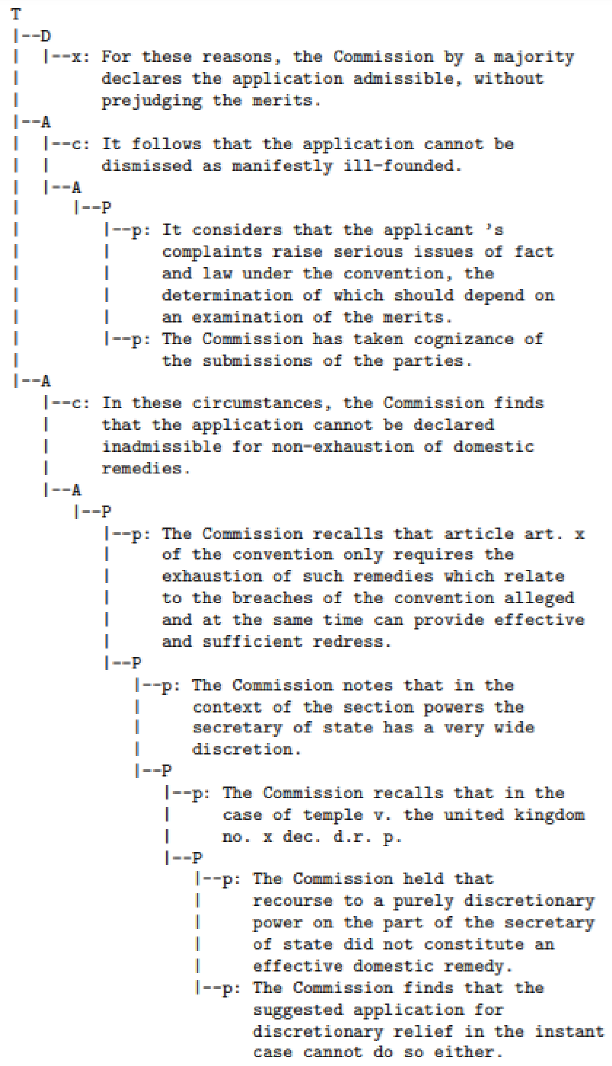
Figure 6: Argumentation Tree Structure

- **Argument Clause recognition**
  Given a clause determine whether it is an argumentative clause or a non-argumentative clause. Can be framed as a binary classification task.

- **Argument Relation Mining**
  Group argument clauses into separate arguments. The framing of the task depends on the method used.

- **Conclusion/Premise Recognition**
  Given a set of argument clauses forming an argument, determine which one of them is the conclusion and which ones are the premises. The framing of the task depends on the method used.

The final output of this task will be a tree containing arguments and sub-arguments at each of its nodes as shown in Figure 7.



```
T
|--D
|  |--x: For these reasons, the Commission by a majority
|        declares the application admissible, without
|        prejudging the merits.
|--A
|  |--c: It follows that the application cannot be
|  |     dismissed as manifestly ill-founded.
|  |--A
|     |--P
|        |--p: It considers that the applicant 's
|        |     complaints raise serious issues of fact
|        |     and law under the convention, the
|        |     determination of which should depend on
|        |     an examination of the merits.
|        |--p: The Commission has taken cognizance of
|              the submissions of the parties.
|
|--A
   |--c: In these circumstances, the Commission finds
   |     that the application cannot be declared
   |     inadmissible for non-exhaustion of domestic
   |     remedies.
   |--A
      |--P
         |--p: The Commission recalls that article art. x
         |     of the convention only requires the
         |     exhaustion of such remedies which relate
         |     to the breaches of the convention alleged
         |     and at the same time can provide effective
         |     and sufficient redress.
         |--P
            |--p: The Commission notes that in the
            |     context of the section powers the
            |     secretary of state has a very wide
            |     discretion.
            |--P
               |--p: The Commission recalls that in the
               |     case of temple v. the united kingdom
               |     no. x dec. d.r. p.
               |--P
                  |--p: The Commission held that
                  |     recourse to a purely discretionary
                  |     power on the part of the secretary
                  |     of state did not constitute an
                  |     effective domestic remedy.
                  |--p: The Commission finds that the
                        suggested application for
                        discretionary relief in the instant
                        case cannot do so either.
```

Figure 7: Sample Output

3.3.3 Argument Clause Recognition

Many techniques have been applied to accomplish this task. Palau and Moens (2009) used a maximum entropy model to classify the clauses.

Poudyal et al. (2020) used RoBERTa (Liu et al., 2019) to classify the clauses. The clause is provided as input and the model outputs a score denoting the probability of the clause being argumentative. Overall accuracy is given in Table 6.

Task	Precision	Recall	F-measure
Argument clause recognition	0.697	0.848	0.765
Argument relations mining	0.502	0.521	0.511
Premise recognition	0.832	0.887	0.859
Conclusion recognition	0.589	0.672	0.628

Table 6: Accuracy scores of RoBERTa (Poudyal et al., 2020)

Poudyal et al. (2020) again used RoBERTa (Liu et al., 2019) for the task of Argument Relation Mining. They frame the task as a binary sentence classification task. Given 2 clauses as input to the model, the model gives a score representing whether the clauses belong to the same argument or not. Accuracy given in Table 6.

Palau and Moens (2009) went with a different approach. They developed a Context Free Grammar for the detection of argument structure. The rules used in the grammar are given in Figure 8. Overall accuracy obtained was 60%.

$$T \Rightarrow A^{+}D$$

$$A \Rightarrow \{A^{+}C\,|A^{*}CnP^{+}|\,Cns\,|A^{*}sr_{c}C|\,P^{+}\}$$

$$D \Rightarrow r_{c}f\,\{v_{c}s\mid\cdot\}^{+}$$

$$P \Rightarrow \{P_{verbP}\,|P_{art}|\,PP_{sup}\,|PP_{ag}|\,sP_{sup}\mid sP_{ag}\}$$

$$P_{verbP} = sv_{p}s$$

$$P_{art} = sr_{art}s$$

$$P_{sup} = \{r_{s}\}\,\{s\,|P_{\text{verbP}}\mid P_{\text{art}}\;|P_{sup}|\,P_{ag}\}$$

$$P_{ag} = \{r_{a}\}\,\{s\,|P_{\text{verbP}}\mid P_{\text{art}}\;|P_{\text{sup}}\mid P_{ag}\}$$

$$C = \{r_{c}\mid r_{s}\}\,\{s|C|r_{c}P_{\text{verbP}}\;\}\;C = s^{*}v_{c}s$$

Figure 8: CFG for Argumentation Parsing (Palau and Moens, 2009)

Palau and Moens (2009) approached the problem of Conclusion/Premise Recognition as a binary classification task. Given a clause/sentence, determine whether it is a premise or a conclusion. Note that the technique was applied on a different legal argumentation mining dataset which is not available publicly. So the actual details might be different, but the overall technique applied is the same. They used 2 Support Vector Machines for this task, one for conclusion detection and the other one is for premise detection. The accuracy scores are provided in Table 3.3.3.

Poudyal et al. (2020) again used Roberta (Liu et al., 2019) for this task. They train 2 models in the same way as Palau and Moens (2009). One important thing to be noted is that it can also be designed as a multi-label classification task where given a group of clauses determine the label of each clause. The accuracy scores are given in Table 6.

4 Summary

This paper discussed the tasks and the techniques which were necessary to understand the work done in for legal assistance. Firstly, it explained in detail the weakly supervised techniques for prior case retrieval. Then, it explained the methods used for rhetorical role classification which combined probabilistic graphical models with deep learning systems to get the benefits of both worlds. Lastly, it also explained the challenging field of legal argumentation mining and explained the newest and oldest approaches implemented for the same.

References

Basit Ali, Ravina More, Sachin Pawar, and Girish Palshikar. 2021. Prior case retrieval using evidence extraction from court judgements. In *Proceedings of the Fifth Workshop on Automated Semantic Analysis of Information in Legal Text*, São Paulo, Brazil. International Association of Artificial Intelligence and Law.

Corina Andone. 2005. A systematic theory of argumentation. the pragma-dialectical approach: Frans h. van eemeren, rob grootendorst, cambridge university press, cambridge, 2004, 216 pages, price £ 15.99 (us22.00)*paperback*, £40.00(*us* 58.00) hardback, isbn 0-521-53772-x paperback, isbn 0-521-83075-3 hardback. *Journal of Pragmatics*, 37(4):577–583.

Paheli Bhattacharya, Shounak Paul, Kripabandhu Ghosh, Saptarshi Ghosh, and Adam Zachary Wyner. 2019. Identification of rhetorical roles of sentences in indian legal judgments. *ArXiv*, abs/1911.05405.

Kripabandhu Ghosh, Sachin Pawar, Girish Keshav Palshikar, Pushpak Bhattacharyya, and Vasudeva verma. 2020. Retrieval of prior court cases using witness testimonies. In *JURIX*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, page 98–107, New York, NY, USA. Association for Computing Machinery.

Prakash Poudyal, Jaromir Savelka, 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, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

M. Saravanan, B. Ravindran, and S. Raman. 2008. Automatic identification of rhetorical roles using conditional random fields for legal document summarization. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.