Prior Case Retrieval using Evidence Extraction from Court Judgements

Basit Ali, Ravina More, Sachin Pawar and Girish K. Palshikar

TCS Research, Pune, India.

Abstract

One of the key constituents of court case descriptions is *Evidence* description and observations. Along with witness testimonies, evidence plays a significant role in the final decision of the case. We propose a weakly supervised technique to automatically identify sentences containing evidences. We represent the information related to evidences in these sentences in a semantically rich structure – *Evidence Structure* defined as an Evidence Information Model. We show that witness testimony information can also be represented using the same model. We demonstrate the effectiveness of our Evidence Information Model for the prior case retrieval application by proposing a matching algorithm for computing semantic similarity between a query and a sentence in a court case description. To the best of our knowledge, this is the first paper to apply NLP techniques for the extraction of evidence information from court judgements and use it for retrieving relevant prior court cases.

Keywords

Evidence Extraction, Evidence Information Model, Natural Language Processing, Prior Case Retrieval

1. Introduction

Evidences - typically based on documents (e.g., letter, receipt, report, agreements, affidavits) and physical objects (e.g., knife, guns, photos, phone call data records) - are often used by lawyers in their arguments during a court case. The observations made through these evidences may have a significant effect on the judges' final decision. In order to develop a deeper understanding of the past court cases, it is valuable to identify various *Evidences* discussed in these cases and the observations which are made about them or through them. Such information about evidences has several applications such as understanding and representing legal arguments, determining strengths and weaknesses of those arguments, identifying relevant past cases in which similar evidences were discussed, etc.

In this paper, we discuss Natural Language Processing (NLP) based techniques for extracting information regarding *Evidences* mentioned in court judgement documents. We propose to represent this information in a rich semantic structure – *Evidence Structure* defined as an Evidence Information Model. Along with *Evidences*, we also identify and represent *Witness Testimonies* using the same Information Model. Initially, we discuss a two-step approach for identifying evidence and testimony sentences. In the first step, linguistic rules are used to determine whether a sentence contains any evidence or testimony information. Here, we use the rules proposed in Ghosh et al. [1] for identification of witness testimonies and design new rules for identification of evidence sentences. In the second step, we train a *Weakly Supervised Sentence Classifier* whose training data is automatically created using the sentences identified by the linguistic rules. It is a multi-label classifier which predicts whether any sentence contains an Evidence or Witness Testimony or both. Once all the Evidence and Testimony sentences are identified from the corpus of court judgements, we propose a Semantic Role Labelling (SRL) [2] based technique to automatically instantiate Evidence Structures for these sentences.

To demonstrate effectiveness of the proposed Evidence Structure, we discuss its use in the prior case retrieval application. We propose a matching algorithm for computing semantic similarity between a query and a sentence in a court judgement document. This algorithm makes use of the proposed Evidence Structure in which both the query and the sentence are represented, resulting in a semantically sound similarity score between them.

Previously, Ghosh et al. [1] identified witness testimonies from court case documents and used them for retrieving relevant prior cases. We propose that considering only witness testimonies leads to loss of key information regarding evidences mentioned in a case. Hence, we identify and use information about various evidences mentioned in the case documents leading to much better prior case retrieval performance as demonstrated in the experiments section. Moreover, Ghosh et al. [1] use a much limited semantic structure to represent information regarding events mentioned in witness testimonies. This structure does not capture important semantic information like whether event is negated, what are the causes behind the event, the manner in which

Proceedings of the Fifth Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2021), June 25, 2021, São Paulo, Brazil.

ali.basit@tcs.com (B. Ali); ravina.m@tcs.com (R. More); sachin7.p@tcs.com (S. Pawar); gk.palshikar@tcs.com (G. K. Palshikar)

event takes place etc. We propose a richer semantic structure addressing these limitations and design a suitable semantic matching algorithm for that structure. To the best of our knowledge, this is the first paper to apply NLP techniques for the extraction of evidence information from court judgements and demonstrate its use for retrieving relevant prior court cases.

2. Evidence Information Model

The purpose of Evidence Information Model is to define a suitable structure to represent evidence information in court judgements. In this section, we describe Semantic Role Labelling in brief and how it is used to define our proposed Evidence Structure.

2.1. Background: Semantic Role Labelling

Semantic Role Labelling (SRL) is a technique in Natural Language Processing that identifies verbs/predicates in a sentence, finds phrases connected to every predicate and assigns an appropriate semantic role to every phrase. By doing so, SRL helps machines to understand the roles of important words within a sentence. Following are some key semantic roles identified for a verb/predicate (often corresponding to an action or event) by SRL techniques:
ARG0 : proto-agent or someone who performs the action denoted by the verb
ARG1 : proto-patient or someone on whom the action is performed on
ARGM-TMP : the time when the event took place
ARGM-CAU : the cause of the action
ARGM-PRP : the purpose of the action
ARGM-LOC : the location where the event took place
ARGM-MNR : the manner in which the action took place
ARGM-NEG : the word indicating that the action did not take place

Consider the following example sentence:
`On August 25, 1965, the bank dishonoured the cheque due to insufficient balance.`

The various semantic roles to the verb `dishonoured` are annotated as follows:
`[ARGM-TMP: On August 25 , 1965] , [ARG0: the bank] [V: dishonoured] [ARG1: the cheque] [ARGM-CAU: due to insufficient balance].`

We use the predicates and corresponding arguments obtained from the pre-trained AllenSRL model [3] to instantiate our Evidence Structure for the queries and candidate sentences.

2.2. Evidence Structure

The *Evidence Information Model* represents every *Evidence Sentence* giving information about one or more *Evidence Objects* in an *Evidence Structure*. We define an *Evidence Object* as one of the objects presented by the counsels to the judge along with the information and findings about the crime. It is thus, a physical entity that can furnish some degree of support, contradiction or opposition to some legal arguments. Some examples of *Evidence Objects* are:

- Documents (`autopsy report`, `post-mortem report`, `affidavit`, `letter`, `cheque`, `agreement`, `petition`, `FIR`, `signature`)
- Material objects (`gun`, `bullet`, `clothes`, `kerosene can`)
- Substances (`poison`, `alcohol`, `kerosene`)

In Indian court case documents, such *Evidence Objects* are also represented in the judgement document as `Exhibit A`, `Ex. 2`, `Evidence 23` and so on.

On these lines, we define an *Evidence Sentence* as any sentence containing one or more *Evidence Objects* relevant to the current case but do not consist of

- any witness testimony which is not verifiable
- legal argumentation
- a reference to some prior case or some Act or Section
- directions or instructions given by the court or judge.

We now present a formal definition of the *Evidence Structure*. For every evidence present in an *Evidence Sentence*, the structure consists of an optional *Observation Frame* and a mandatory *Evidence Frame*. The *Observation Frame* represents the source of the information and the agent disclosing it. This information is optional as it may or may not be explicitly stated in a sentence. It consists of the following arguments:

- **ObserverVerb or OV**: The verb indicating the observation/discovery/disclosure (e.g., `found`, `revealed`, `stated`)
- **ObserverAgent or A_0**: 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* captures details about the evidence itself through the following arguments:

- **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 A_0**: someone who initiates the action indicated by the EvidenceVerb (e.g., `the accused`, `Ram`, `ABC Pvt. Ltd.`)
- **Patient or A_1**: someone who undergoes the action indicated by the EvidenceVerb. (e.g., `the`

Table 1
Example Evidence sentences with their Evidence Structure Instances

The bank dishonoured the cheque due to insufficient balance. • EF = [EV = dishonoured, A_0 = The bank, A_1 = the cheque, CAU = due to insufficient balance]
The report revealed that organo-phosphorus compound was found in the stomach , small intestines , large intestines , liver , spleen , kidney and brain of the deceased . • OF = [OV = revealed, EO = The report] EF = [EV = found, LOC = in the stomach , small intestines , large intestines , liver , spleen , kidney and brain of the deceased]
The Magistrate found prima facie evidence that the appellant had fraudulently used in the Civil Suit forged cheque and committed him to the Sessions for trial • OF = [OV = found, OA = The Magistrate, EO = prima facie evidence] EF = [EV = used,A_0 = the appellant, A_1 = forged cheque, LOC = in the Civil Suit]
The prosecution case was that though the rough cash book showed that on September 29, 1950 a sum of Rs. 21,133 was sent to the Treasury by appellant Gupta , the Treasury figures in the challan showed that on that day only a sum of Rs. 1,133 was deposited into the Treasury and thus a sum of Rs.20,000 was dishonestly misappropriated . • OF = [OV = showed, EO = the rough cash book] EF = [EV = sent, A_0 = by appellant Gupta, A_1 = a sum of Rs.21,133, A_2 = to the Treasury, $ARG{-}TMP$ = on September 29,1950] • OF = [OV = showed, EO = the Treasury figures in the challan] EF = [EV = deposited,A_0 = by appellant Gupta, A_1 = only a sum of Rs.1,133, A_2 = into the Treasury, TMP = on that day] • OF = [OV = showed, EO = the Treasury figures in the challan] EF = [EV = misappropriated, A_1 = a sum of Rs.20,000, MNR = dishonestly]

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

Table 1 shows examples of some Evidence Sentences along with the corresponding *Evidence Structure Instances*. In some cases, Observation Frame may be empty due to absence of ObservationVerb. In such cases, EvidenceObject may be present as a part of any argument in Evidence Frame. E.g., the cheque is present as A_1 in the Evidence Frame of the first sentence in Table 1.

Information about named entities and their types present in various arguments of Observation or Evidence frame is important. Hence, the *Observation Frame* and *Evidence Frame* are also enriched by annotating entities such as PERSON, ORGANISATION, GEO-POLITICAL ENTITY, LOCATION, PRODUCT, EVENT, LANGUAGE, DATE, TIME, PERCENT, MONEY, QUANTITY, ORDINAL, CARDINAL, WEAPON, SUBSTANCE, DOCUMENT, ARTIFACT, WORK_OF_ART, WITNESS, BODY_PART, and VEHICLE present in the fields.

Witness Information Model: Information in witness testimonies can also be represented using the same *Evidence Structure*. The statement verbs used in witness testimony sentences (e.g., stated, said) are treated similar to observation verbs and represented using Observation Frames. Similarly, other action/event verbs mentioned in witness testimony sentences are represented using Evidence Frames. Table 2 shows examples of some *Witness Sentences* along with the corresponding *Evidence Structure Instances*. The advantage of representing information about evidences and witness testimonies in the same structure is that we can make use of both these sources of information seamlessly, for prior case retrieval.

3. Methodology

In this section, we describe our overall methodology which consists of two phases. In the first phase, we identify *Evidence and Testimony Sentences* using linguistic rules and weakly supervised sentence classifier. In the second phase, we instantiate the Evidence Structures for these identified sentences. For all our experiments, we use a corpus of 30,032 Indian Supreme Court judgements ranging from the year 1952 to 2012.

3.1. Identification of Evidence and Testimony Sentences

We identify Evidence and Testimony sentences using a two-step approach. In the first step, we use linguistic rules to obtain Evidence and Testimony sentences. In the second step, we use these sentences to train a sentence classifier.

Table 2
Example Witness Testimony sentences with their Evidence Structure Instances

He has categorically stated that by reason of enmity , A1 and A2 together have murdered his brother-in-law . • OF = [OV = stated, A_0 = He] EF = [EV = murdered, A_0 = A1 and A2 together, A_1 = his brother-in-law, CAU = by reason of enmity]
Shri Dholey (PW-6) reiterated about the dacoity and claimed that a pistol was brandished on him by one of the accused persons . • OF = [OV = claimed, A_0 = Shri Dholey (PW-6)] EF = [EV = brandished, A_0 = by one of the accused persons, A_1 = a pistol, CAU = on him]
Though he stated in the post-mortem report that death would have occurred about 12 hours back , he clarified that there was possibility of injuries being received at about 9 A.M. • OF = [OV = stated, A_0 = he, EO = the post-mortem report] EF = [EV = occurred, A_1 = death, TMP = about 12 hours back] • EF = [OV = clarified, A_0 = he, A_1 = that there was possibility of injuries being received at about 9 A.M. Deceased Sarit Khanna was aged about 27 years]
He admitted , however , that Shri Buch had met him in connection with the covenant , but he denied that he had received any letter Exhibit P-9 from Shri Buch or the lists Exhibits P- 10 to P- 12 regarding his private and State properties , were a part thereof . • OF = [OV = admitted, A_0 = He] EF = [EV = met, A_0 = Shri Buch, A_1 = him, TMP = in connection with the covenant] • OF = [OV = denied, A_0 = He] EF = [EV = received, A_0 = he, A_1 = any letter Exhibit P-9, A_2 = from Shri Buch]

Step I: Linguistic Rules based Approach: As there are no publicly annotated datasets for identification of Evidence and Testimony sentences, we rely on linguistic rules to identify these sentences with high precision as our first step. The linguistic rules for identifying Evidence sentences are described in detail in Table 3. These rules identified 62,310 sentences as Evidences from our corpus. As there is no annotated dataset, in order to estimate the precision of the linguistic rules we use random sampling strategy. We selected a set 100 random sentences identified as Evidence by the linguistic rule, and got them verified by a human expert. The precision turned out to be 85%. Similarly, we use the linguistic rules proposed in Ghosh et al. [1] for identifying Testimony and non-Testimony sentences where the reported precision is around 85%. These rules identified 36,473 sentence as Testimony and 14,234 sentences as non-Testimony from the same corpus.

Step II: Weakly Supervised Sentence Classification: We observed that although the linguistic rules identify Evidence and Testimony sentences with high precision, they may miss to identify some sentences which should have been identified as Evidence or Testimony (see examples in Table 4). Hence, we train a supervised sentence classifier to improve overall recall of identification of Evidence and Testimony sentences. The classifier used is a BiLSTM-based [4] multi-label sentence classifier whose architecture is depicted in Figure 1. This classifier is weakly supervised since its training data is automatically created using the sentences identified by the linguistic rules as follows:

• The classifier has two outputs - i) first output predicts a binary label indicating whether the sentence contains Evidence or not and ii) second output predicts a binary label indicating whether the sentence contains Testimony or not.
• 1824 sentences are labelled as Evidence and Testimony both. These sentences are identified as Evidence as well as Testimony by both the sets of linguistic rules.
• 60486 sentences are labelled as Evidence and non-Testimony. These sentences are identified as Evidence by the rules but not as Testimony.
• 34649 sentences are labelled as non-Evidence and Testimony. These sentences are identified as Testimony by the rules but not as Evidence.
• 14234 sentences are labelled as non-Evidence and non-Testimony. These sentences are identified as non-Testimony by the rules and not identified as Evidence.

After this classifier is trained, we use it to classify all the remaining sentences in the corpus. These sentences are neither identified Evidence by the Evidence rules nor as Testimony/non-Testimony by the Testimony rules. Using the prediction confidence, we selected top 10,000 sentences classified as Evidence and top 5,000 sentences classified as Testimony. Table 4 shows some examples of sentences identified as Evidence by the classifier but not by the linguistic rules. To estimate the precision, we again employed the random sampling strategy. We selected 100 random sentences each from these high confidence Evidence and Testimony sentences and a human expert verified them. The precision of 72% is observed for Evidence sentences and 68% for Testimony sentences. The precision of the sentence classifier is lower as compared

Table 3
Linguistic Rules for identifying Evidence Sentences

Any sentence *S* should satisfy the following conditions in order to be identified as an Evidence Sentence:	
E-R1	*S* should contain at least one *Evidence Object* as defined in Section 2.2. The list of words corresponding to evidence objects is created automatically by using WordNet hypernym structure. We create a list of 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`). This list is looked up to identify evidence objects in a sentence.
E-R2	*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.
E-R3	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.

Table 4
Example of Evidence Sentences Identified by the Classifier but not by the linguistic rules

`S1: Raju PW2 took Preeti into the bath room at the instance of Accused No.1 who cut a length of wire of washing machine and used it to choke her to death, who however, survived.`
`S2: Raju PW2 took Satyabhamabai Sutar in the kitchen where the accused No.1 had already reached and was washing the blood stained knife.`
`S3: Hemlata was also killed by inflicting knife injuries.`
`S4: Accused No.2 and Raju PW2 took the child into the room where Meerabai was lying dead in the pool of blood.`
`S5: Accused No.2 gave her blows by putting his knees on her stomach and when she was immobilised this way , the Accused No.1 gave her knife blows on her neck with the result she also died.`
`S6: Almirahs found in the flat were emptied to the extent the accused could put articles and other cash and valuables in the air-bag obtained from the said flat.`
`S7: Blood stained clothes of Accused No.2 were put in the air-bag along with stolen articles.`

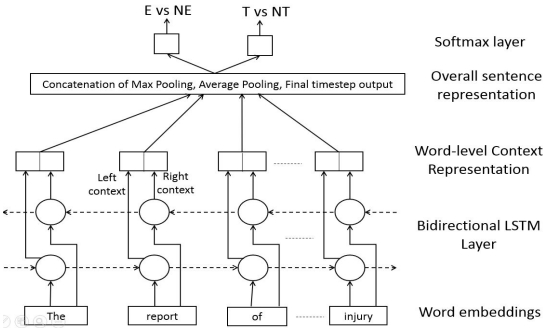

Figure 1: Architecture of the BiLSTM-based multi-label sentence classifier (T: Testimony, NT: Non-Testimony, E: Evidence, NE: Non-Evidence)

to the rules because, it is applied on a more *difficult* set of sentences for which the linguistic rules fail to identify any label. At the end of this two-step process (linguistic rules followed by the sentence classifier), we have 112,401 sentences identified either as Evidence or as Testimony.

3.2. Evidence Structure Instances

In this phase, we discuss the technique of instantiating Evidence Structures for sentences identified as Evidence or Testimony in the previous phase. We used Semantic Role Labelling [3] to identify and fill the arguments of the Observation Frame and the Evidence Frame in the Evidence Structure Instance for every candidate sentence. This is demonstrated in Algorithm 1. We identify *Observation Frames* using Observation Cue Verbs. For each of these *Observation Frames* we identify the corresponding *Evidence Objects* and *Evidence Frames*. For identifying *Evidence Objects*, we first use Named Entity Recognition [5] and WordNet based Entity Identification [6] to identify the named entities in the sentence and annotate them in the Frames extracted. The *Evidence Objects* in a phrase are then obtained by selecting named entities annotated as one of the following types - `ARTIFACT`, `VEHICLE`, `WEAPON`, `DOCUMENT`, `WORK_OF_ART`, `SUBSTANCE`. This corresponds to the *get_evidence_object* function used in Algorithm 1. *Observation Frames* that do not contain a corresponding *Evidence Frame* are redesigned as stand alone *Evidence Frames*. We finally combine the Evidence Frame and the Observation Frame into an Evidence Structure Instance.

We measured the accuracy of 260 *Evidence Structure*

Instances obtained from 100 random Evidence and Testimony sentences. The accuracy of the Observation Frame extraction is 86% and that of Evidence Frame extraction is 88%. We observed that most of the incorrect extractions were due to parsing error in the SRL model.

4. Prior Case Retrieval

In order to demonstrate effectiveness of the proposed Evidence Structure, we apply it for the task of prior case retrieval. This task is to create a relevance-based ranked list of court judgements (documents) in our corpus for a query. In order to retrieve prior cases for a query, we represent the query using an Evidence Structure Instance ($EvStruct_Q$). We then compute the similarity of query instance $EvStruct_Q$ against each document instance $EvStruct_D$ obtained from every Evidence or Testimony sentence in the corpus. Algorithm 2 shows the steps for computing similarity. We refer to this algorithm as *SemMatch* because of its *semantic matching* ability. We use cosine similarity between the phrase embeddings of corresponding arguments of the Evidence Structure Instances to compute similarity. For obtaining phrase embedding for any phrase (referred as *PhraseVec* in Algorithm 2), we consider the average of GloVe word embeddings [7] of the words in that phrase excluding stop words. We compute the similarity scores within corresponding arguments of both the frames. These scores across different arguments are combined to get a final similarity score between $EvStruct_Q$ and $EvStruct_D$. We multiply the final similarity score by a Sentence BERT [8] based similarity score between the query and the sentence containing $EvStruct_D$. This is necessary because errors in the automated SRL tool may lead to imperfect Evidence Structure instances in some cases. A sentence similarity score which is not dependent on any such structure within the sentences provides a complementary view of capturing sentence similarity. Finally, the overall relevance score of the query with a document is the maximum score corresponding to any Evidence Structure Instance $EvStruct_D$ obtained from the document. Table 5 shows a running example of how a similarity score is computed between an Evidence Structure Instance ($EvStruct_Q$) from a query and an Evidence Structure Instance ($EvStruct_D$) from a document in the corpus.

5. Related Work

While the task of evidence extraction from legal documents is related to several information retrieval and NLP tasks, there are no established baselines for the task. Bellot et al. [9] and Cartright et al. [10] have worked on Evidence Retrieval that identifies whole documents that contain an evidence. On the other hand, Rinott et al. [11] use *Context Dependent Evidence Detection* to find evidence information present in a sentence on a phrase level. As compared to this, we identify both Evidence and Testimony sentences, represent them in a rich structure and also use that for prior case retrieval. This is a challenging task due to the inherently complex nature of legal texts and the finer granularity of matching involved.

Ji et al. [12] propose an Evidence Information Extraction system which captures evidence production paragraph, evidence cross-examination paragraph, evidence provider, evidence name, evidence content, cross-examination party and cross-examination opinion relating to an evidence presented in the court. While this technique may suit well for Chinese court records that follow a relatively structured representation, it does not suit well to the Indian Court Records that contain descriptive and varied formats of the court proceedings.

Gomes and Ladeira [13] and Landthaler et al. [14] performs full text search for legal document collection by obtaining word2vec word embeddings and then taking their average for computing similarity. However, computing the average of the embeddings gives a lossy representation where relative order of the words is lost. In contrast, we represent the sentences using the Evidence Structure Instances, where the structure itself takes care of the relative ordering. Gomes and Ladeira [13] demonstrate BM25 and TF-IDF for Prior Case retrieval. In our results section, we demonstrate the comparative poor performance of BM25 and TF-IDF in handling corner cases.

6. Experimental Evaluation

In this section, we discuss our experiments including the dataset, baseline techniques, evaluation metrics and analysis of results.

6.1. Dataset

We use the Indian Supreme Court judgements from years 1952 to 2012 freely available at http://liiofindia.org/in/cases/cen/INSC/. There are 30032 court (documents) containing 4,111,091 sentences where average sentence length is 31 words and standard deviation of 24.

6.2. Baselines

For the task of prior case retrieval, we implement two baseline techniques:

• **BM25**: It is a popular TF-IDF based relevance computation technique. We use the BM25+ variant[1] as described in Trotman et al. [15]. This technique uses a bag-of-words approach that ignores the sentence structure. We

[1] https://pypi.org/project/rank-bm25/

input : s (sentence), SRL_P (set of semantic frames in s as per any semantic role labeller, each frame P consists of a predicate $P.V$ and corresponding arguments $P.ARG_0$, $P.ARG_1$, $P.ARG_2$, $P.ARGM\text{-}LOC$, etc.)
output : $EvStructs$ = Evidence Structure Instances of the input sentence consisting of *ObservationFrame* (*OF*) and *EvidenceFrame* (*EF*)
parameter : OBS_VERBS = {accept, add, admit, agree, allege, allow, alter, apprise, assert, brief, build, challenge, claim, clarify, complain, confirm, corroborate, decline, demand, deny, depose, describe, disclose, dismiss, examine, exhibit, find, include, indicate, inform, mention, note, notice, observe, obtain, occur, point, prepare, present, receive, recover, refuse, reject, remember, report, reveal, say, show, state, submit, suggest, tell, withdraw}, NEG_WORDS = {no, not, neither, nor, never}

$EvStructs := \emptyset$
$OFs := \emptyset$
// Obtain Observation Frames in the sentence s
foreach $P \in SRL\_P$ *such that* $P.V \in OBS\_VERBS$ **do**
  $OF :=$ Create empty Observation Frame
  $OF.V := P.V$
  $OF.NEG := P.ARGM\text{-}NEG$
  $OF.A_0 := P.ARG_0$
  $OF.A_1 := P.ARG_1$
  // If any of the arguments of the predicate starts with a negative word, then we negate the verb.
  if $OF.A_0$ *or* $OF.A_1$ *starts with any word from* NEG_WORDS **then**
    $OF.NEG := True$
  $OF.EO := get\_evidence\_object(P.ARG_0) \cup get\_evidence\_object(P.ARGM\text{-}LOC)$
  $OFs := OFs \cup \{OF\}$

// Obtain corresponding Evidence Frames for every Observation Frame
foreach $OF \in OFs$ **do**
  $FoundEF := False$
  foreach $P \in SRL\_P$ *such that* $P.V$ *occurs within the span of* $OF.A_1$ **do**
    if $P.V$ *is a copula verb and any of* $P.ARG_0$ *or* $P.ARG_1$ *does not exist* **then**
      continue
    $EF :=$ Create empty Evidence Frame
    $EF.V := P.V$
    $EF.NEG := P.ARGM\text{-}NEG$
    // If any of the arguments of the predicate starts with a negative word, then we negate the verb
    if $OF.A_0$ *or* $OF.A_1$ *starts with any word from* NEG_WORDS **then**
      $EF.NEG := True$
    foreach *argument* $ARG \in P.arguments$ **do**
      $EF.ARG := P.ARG$
    delete($OF.A_1$)
    $EvStruct := \{(OF, EF)\}$
    $EvStructs := EvStructs \cup EvStruct$
    $FoundEF := True$

  // If no Evidence Frame exists for an Observation Frame, transfer the Observation Frame to the Evidence Frame
  if $FoundEF == False$ **then**
    $EF :=$ Create empty Evidence Frame
    $EF.V := OF.V$
    $P := P' \in SRL\_P$ such that $P'.V = OF.V$
    foreach *argument* $ARG \in P.arguments$ **do**
      $EF.ARG := P.ARG$
    clear(OF)
    $OF.EO := get\_evidence\_object(P.ARG_0) \cup get\_evidence\_object(P.ARGM\text{-}LOC)$
    //Add all the required arguments to Evidence Frame
    $EvStruct := \{(OF, EF)\}$
    $EvStructs := EvStructs \cup EvStruct$
return($EvStructs$)

Algorithm 1: *get_evidence_structure_instances*: Algorithm for instantiating Evidence Structure for a sentence

input : $EvStruct_Q$: Evidence Structure Instance from a query sentence Q
$EvStruct_D$: Evidence Structure Instance from a sentence D in the corpus
output: Similarity score between $EvStruct_Q$ and $EvStruct_D$

// Checking for negation
if $EvStruct_Q.OF.NEG \neq EvStruct_D.OF.NEG$ **then return** 0

if $EvStruct_Q.EF.NEG \neq EvStruct_D.EF.NEG$ **then return** 0

// Computing similarity between main predicates, using cosine similarity of their word embeddings
$sim_E := CosineSim(WordVec(EvStruct_Q.EF.V), WordVec(EvStruct_D.EF.V))$

// Computing similarity between corresponding Evidence Objects, using cosine similarity of their phrase embeddings
$sim_{EO} := CosineSim(PhraseVec(EvStruct_Q.OF.EO), PhraseVec(EvStruct_D.OF.EO))$

// Computing similarity between other arguments, using cosine similarity of their phrase embeddings
$num_{args} := 0$
$sim_{args} := 0$
foreach $arg \in (EvStruct_Q.EF.arguments - \{V\})$ **do**
  if $EvStruct_Q.EF.arg$ *exists* **then**
    $sim_{args} := sim_{args} + CosineSim(PhraseVec(EvStruct_Q.EF.arg), PhraseVec(EvStruct_D.EF.arg))$
    $num_{args} := num_{args} + 1$
$sim_{args} := sim_{args}/num_{args}$

// Computing overall similarity
$sim_{final} := sim_E \times sim_{args} \times sim_{EO}$

// The overall similarity is multiplied by the Sentence-BERT based sentence similarity between Q and D
$sim_{final} := sim_{final} \times CosineSim(SentVec(Q), SentVec(D))$

return sim_{final}

Algorithm 2: *SemMatch*: Algorithm for computing similarity between $EvStruct_Q$ and $EvStruct_D$

Table 5
Example of the proposed *SemMatch* algorithm in action

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 6
Evaluation of various techniques for the task of prior case retrieval. All entries are of the form (R-Prec; Avg. Precision). (Note: Our proposed approach *SemMatch* is referred as *SM*. Underlines indicate the best performing results for each query across multiple techniques)

What are the cases where⋯
Q_1: blood stains were found on 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 the exhibited gun was not used.
Q_5: the autopsy report reveals that some poisonous compounds are 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 police.
Q_{10}: a signature was forged on an affidavit.

Query	$BM25_{all}$	$BM25_T$	$BM25_E$	$BM25_{TE}$	SB_T	SB_E	SB_{TE}	SM_T	SM_E	SM_{TE}
Q_1	0.24; 0.26	0.06; 0.02	0.59; 0.49	0.59; 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; 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	0.50; 0.30
Q_3	0.00; 0.01	0.00; 0.03	0.33; 0.33	0.33; 0.35	0.33; 0.12	0.00; 0.00	0.00; 0.09	0.33; 0.12	0.00; 0.00	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	0.70; 0.80	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	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	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	0.71; 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	0.50; 0.51	0.40; 0.41	0.10; 0.12	0.50; 0.48
Q_{10}	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	0.50; 0.61	0.50; 0.61
Avg	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	0.40; 0.35

use 4 settings considering different sentences in each document:
• $BM25_{all}$: All sentences
• $BM25_{TE}$: Only Testimony or Evidence sentences
• $BM25_T$: Only Testimony sentences
• $BM25_E$: Only Evidence sentences
• **Sentence-BERT** [8]: This technique is based on Siamese-BERT networks to obtain more meaningful sentence embeddings as compared to vanilla BERT [16]. We used the pre-trained model `bert-base-nli-stsb-mean-tokens` to obtain sentence embeddings for sentences. Following Ghosh et al. [1], we use the pre-trained model as it is and did not fine-tune it further. This is because such fine-tuning needs annotated sentence pairs with labels indicating whether the sentences in the pair are semantically similar or not. Such annotated dataset is expensive to create and our aim is to avoid any dependence on manually annotated training data. Similar to Ghosh et al. [1], we used sentence embeddings obtained by Sentence-BERT to compute cosine similarity between a query sentence and a candidate sentence in a document. The overall similarity of a document with a query is the maximum cosine similarity obtained for any of its sentences with the query sentence. We use 3 settings considering different sentences in each document:
• SB_{TE}: Only Testimony or Evidence sentences
• SB_T: Only Testimony sentences
• SB_E: Only Evidence sentences

6.3. Evaluation

All the baseline techniques and our proposed technique are evaluated using a set of queries and using certain evaluation metrics to evaluate and compare the ranked lists produced by each of these techniques.
Queries: We chose 10 queries (shown in Table 6) which represent cases and evidence objects of diverse nature (domestic violence, financial fraud etc.).
Ground Truth: We created a set of gold-standard relevant documents for each query using the standard *pooling technique* [17]. We ran the following techniques to produce a ranked list of documents for each query – $BM25_{all}$, $BM25_{TE}$, SB_{TE}, and our proposed technique $SemMatch_{TE}$. We chose top 10 documents from the ranked list produced by each technique. Human experts verified the relevance of each document for the query. Finally, after discarding all the irrelevant documents, we got a set of gold-standard relevant documents for each query[2].
Metrics: We used R-Precision and Average Precision as our evaluation metrics [17].

1. **R-Precision (R-Prec)**: This calculates the the number of relevant documents observed at R.
2. **Average Precision (AP)**: This captures the joint effect of Precision and Recall. It computes precision at each rank of the predicted ranked list and then computes mean of these precision values.

[2]This dataset can be obtained from the authors on request

6.4. Results

Table 6 shows comparative evaluation results for various baselines and our proposed technique. Average performance of $BM25_{TE}$ is better than $BM25_{all}$ indicating that considering only Evidence and Testimony sentences for representing any document, results in better prior case retrieval performance. Other two baselines *SB* (Sentence-BERT) and *SM* (our proposed technique *SemMatch*) also consider only Evidence and Testimony sentences rather than considering all the sentences in a document. All the baselines which consider only Testimony sentences, perform poorly as compared to the corresponding techniques using both Testimony and Evidence sentences. This highlights the importance of evidence information as compared to using only witness testimony information for prior case retrieval as done in Ghosh et al. [1].

Considering the average performance across all the 10 queries, our proposed technique SM_{TE} is the best performing technique in terms of both R-Prec and AP. The performance of SM_{TE} is the most consistent across the diverse queries. It achieves minimum R-Prec of 0.24 (for Q_1) as compared to other baselines like $BM25_{all}$, $BM25_{TE}$ and SB_{TE} which have minimum R-Prec of 0 for some queries. As described in Algorithm 2, *SM* uses Sentence-BERT based similarity within sentences for producing an enhanced matching score. We experimented with a variant of *SM* which does not rely on Sentence-BERT based similarity. This variant resulted in average R-Prec of 0.36 and MAP of 0.30 across all the 10 queries. Although this is lower than SM_{TE} performance, the R-Prec is still comparable with $BM25_{TE}$ (avg R-Prec of 0.36) and better than that of SB_{TE} (avg R-Prec of 0.28).

For some queries, it is important to have some semantic understanding at sentence-level. For example, Q_4, which contains "negation", *SB* and *SM* can capture the query's meaning in a better way. *SM* handles such negations in a more principled manner as the Evidence Structure Instance captures negation as one of its arguments.

For *SM*, the maximum matching score achieved for any Evidence Structure Instance in a document, is considered as the overall matching score with the whole document. In contrast, *BM25* based techniques directly compute matching score for the whole document as they do not rely on sentence structure. This is one limitation of *SM* which we plan to address as a future work. However, as *SM* computes matching scores for individual Evidence Structure instances, it is able to provide better interpretation for each relevant document in terms of the actual sentences which provided the maximum matching score.

Analysis of errors: We analyzed cases where SM_{TE} was assigned a lower score to a relevant document or a higher score to a non-relevant document. We discovered 3 main reasons - missing or incorrect arguments within Evidence Structure instances, misleading high similarity between argument phrases and presence of co-references. Consider the following sentence for which SM_{TE} incorrectly assigns a high score for query Q_5 (see Table 6) – `The police report also reveals that three pieces of pellets were found by the doctor in the body of deceased Monu.` Here, except the A_1 argument (`some poisonous compounds` vs `three pieces of pellets`) in Evidence Structure instances, other arguments are similar in meaning. We get cosine similarity of 0.36 between `some poisonous compounds` and `three pieces of pellets` which is misleading. It is not too low as compared to another case where there are semantically similar argument phrases (e.g., cosine similarity between `some poisonous compounds` and `a heavy concentration of arsenic` is just 0.55 as shown in Table 5). As we are not resolving co-references, we are missing a few relevant documents. E.g., SM_{TE} does not assign a high score for the following document for query Q_3 (see Table 6) – `Instead of surrendering before the police, the deceased had attempted to kill the police. In retaliation, he was shot by them.`. This is because `them` in the Evidence Structure instance for `shot` is not explicitly known to correspond to `the police` in the previous sentence.

7. Conclusion and Future Work

In this paper, we discussed several NLP techniques for identifying evidence sentences, representing them in the semantically rich Evidence Structure and retrieving relevant prior cases by exploiting it. The proposed techniques are weakly supervised as they do not rely on any manually annotated training data, except for the human expertise in designing the linguistic rules. Keeping in mind the importance of witness testimonies in addition to evidences, we also extracted and represented the witness testimonies using the same Evidence Structure. For the application of prior case retrieval, we evaluated our proposed technique along with several competent baselines, on a dataset of 10 diverse queries. We demonstrated that our technique performs comparably for most of the queries and is the best considering the overall performance across all 10 queries. The results highlight the contribution of evidence and testimony information in improving prior case retrieval performance.

In future, we plan to apply advanced representation learning techniques for learning dense or embedded representation of an entire Evidence Structure instance. Also, we plan to automatically determine the best suited retrieval technique (BM25, Sentence-BERT or SemMatch) for any query based on its nature. We plan to explore ensemble of multiple retrieval techniques for improving prior case retrieval performance further.

References

[1] K. Ghosh, S. Pawar, G. Palshikar, P. Bhattacharyya, V. Varma, Retrieval of prior court cases using witness testimonies, JURIX (2020).

[2] M. Palmer, D. Gildea, P. Kingsbury, The proposition bank: An annotated corpus of semantic roles, Computational linguistics 31 (2005) 71–106.

[3] P. Shi, J. Lin, Simple BERT models for relation extraction and semantic role labeling, CoRR abs/1904.05255 (2019). URL: http://arxiv.org/abs/1904.05255. arXiv:1904.05255.

[4] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[5] M. Honnibal, I. Montani, S. Van Landeghem, A. Boyd, spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL: https://doi.org/10.5281/zenodo.1212303. doi:10.5281/zenodo.1212303.

[6] G. A. Miller, Wordnet: a lexical database for english, Communications of the ACM 38 (1995) 39–41.

[7] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[8] N. Reimers, I. Gurevych, 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), 2019, pp. 3973–3983.

[9] P. Bellot, A. Doucet, S. Geva, S. Gurajada, J. Kamps, G. Kazai, M. Koolen, A. Mishra, V. Moriceau, J. Mothe, M. Preminger, E. SanJuan, R. Schenkel, X. Tannier, M. Theobald, M. Trappett, Q. Wang, Overview of INEX 2013, in: P. Forner, H. Müller, R. Paredes, P. Rosso, B. Stein (Eds.), Information Access Evaluation. Multilinguality, Multimodality, and Visualization - 4th International Conference of the CLEF Initiative, CLEF 2013, Valencia, Spain, September 23-26, 2013. Proceedings, volume 8138 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 269–281. URL: https://doi.org/10.1007/978-3-642-40802-1_27. doi:10.1007/978-3-642-40802-1_27.

[10] M.-A. Cartright, H. A. Feild, J. Allan, Evidence finding using a collection of books, in: Proceedings of the 4th ACM workshop on Online books, complementary social media and crowdsourcing, 2011, pp. 11–18.

[11] R. Rinott, L. Dankin, C. Alzate, M. M. Khapra, E. Aharoni, N. Slonim, Show me your evidence-an automatic method for context dependent evidence detection, in: Proceedings of the 2015 conference on empirical methods in natural language processing, 2015, pp. 440–450.

[12] D. Ji, P. Tao, H. Fei, Y. Ren, An end-to-end joint model for evidence information extraction from court record document, Information Processing & Management 57 (2020) 102305.

[13] T. Gomes, M. Ladeira, A new conceptual framework for enhancing legal information retrieval at the brazilian superior court of justice, in: Proceedings of the 12th International Conference on Management of Digital EcoSystems, 2020, pp. 26–29.

[14] J. Landthaler, B. Waltl, P. Holl, F. Matthes, Extending full text search for legal document collections using word embeddings., in: JURIX, 2016, pp. 73–82.

[15] A. Trotman, A. Puurula, B. Burgess, Improvements to bm25 and language models examined, in: Proceedings of the 2014 Australasian Document Computing Symposium, 2014, pp. 58–65.

[16] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[17] C. Manning, P. Raghavan, H. Schutze, Introduction to information retrieval, Natural Language Engineering 16 (2010) 100–103.