Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200848

Retrieval of Prior Court Cases Using Witness Testimonies

Kripabandhu GHOSH [b,1], Sachin PAWAR [a], Girish PALSHIKAR [a], Pushpak BHATTACHARYYA [c] and Vasudeva VARMA [d]

[a] *TCS Research, Tata Consultancy Services, Pune, India*
[b] *Indian Institute of Science Education and Research - Kolkata*
[c] *Indian Institute of Technology Bombay*
[d] *International Institute of Information Technology Hyderabad*

Abstract. Witness testimonies are important constituents of a court case description and play a significant role in the final decision. We propose two techniques to identify sentences representing witness testimonies. The first technique employs linguistic rules whereas the second technique applies distant supervision where training set is constructed automatically using the output of the first technique. We then represent the identified witness testimonies in a more meaningful structure – event verb (predicate) along with its arguments corresponding to semantic roles A0 and A1 [1]. We demonstrate effectiveness of such representation in retrieving semantically similar prior relevant cases. To the best of our knowledge, this is the first paper to apply NLP techniques to extract witness information from court judgements and use it for retrieving prior court cases.

Keywords. Prior Case Retrieval, Witness Testimonies, Natural Language Processing, Semantic Roles

1. Introduction

Witnesses – whether prosecution or defence, lay or expert – are important in all types of court cases. Witness testimonies and their cross-examinations by the counsels have a significant effect on the judges' decision. Large corpora of court judgements (e.g., the Indian Supreme and High Court judgements), often contain the judges' summaries of the witness testimonies presented during the proceedings. In addition, judges often comment in the judgement 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 strengths and weaknesses of a case, predicting court decisions and extracting legal argumentation.

In this paper, first, we propose two NLP techniques (linguistic knowledge-based and distantly supervised) to identify sentences of class Testimony, i.e., sentences containing

[1] Work was carried out when the author was in TCS Research, Tata Consultancy Services, Pune

testimonies of witnesses; example: `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.` Further, we extract details of events mentioned in such witness Testimony sentences. A witness testimony provides factual or subjective details about various events, objects and persons. We extract information provided by witnesses about various events, and not about the persons or objects, though the approach can be easily extended. We focus on two types of events: *crime events* or *legal events* such as filing a police complaint, or arrest. We restrict an *event* to mean a physical action or communication. We focus on events expressed as verbs, although nouns (e.g., `attack`) can also denote an event. We represent event information provided by a witness as an *event frame*, consisting of (i) the action verb, (ii) the agent who initiated the action, and (iii) the patient (or beneficiary) who experienced the action. Other event details (e.g., time, location), can be easily extracted. We use MatePlus [2], a semantic role labeling (SRL) tool, to identify the predicate and associated A0, A1 arguments (described in Section 2.2.1) and fill up event frames.

Finally, after extracting the event details from witness testimonies, we demonstrate its use for improving retrieval of relevant past cases (*prior cases*) based on high-level English queries which might be asked by a lawyer or a lay person. *Prior cases* form the backbone of judicial systems following Common Law; e.g., in India. For *prior case retrieval*, we propose two techniques. The first is based on exact matching of the event frames (one from the query and another from a past court judgement). The other is based on learning a representation for event frames and then using a similarity measure over event frames. We demonstrate that our approaches perform better in retrieving past court judgements as compared to three baseline methods: BoW retrieval function BM25, similarity over document representation vectors given by Doc2Vec, and Sentence-BERT. Doc2Vec has demonstrated efficacy in prior case retrieval [3] when the whole case document is considered. However, explainability of a prior case retrieval result remains an open question, which we attempt to address in this paper. Recently, [4] used supervised techniques for answering basic questions in legal domain using numerous features. Our proposed technique for prior case retrieval is completely unsupervised which handles fine-grained questions pertinent to a given case situation. To the best of our knowledge, this is the first paper to apply NLP techniques to extract witness information from court judgements and use it for assisting lawyers in an explainable manner.

2. Methodology

Our proposed technique works in two phases. In the first phase, we identify witness testimony sentences from prior court case documents using a set of linguistic rules and a distantly supervised LSTM-based sentence classifier. Then in the second phase, we retrieve prior court cases relevant to a query using two different matching techniques. Here, the queries are matched with only witness testimony sentences from the prior court cases and other sentences are ignored. We use a corpus of 30,034 Indian Supreme Court judgements from years 1952 to 2012, for all our experiments.

2.1. Identifying Witness Testimony Sentences

As there are no readily available annotated datasets for Testimony sentences, we use linguistic rules to create training data automatically. This technique works in two steps

where in the first step, the *linguistic rules* are used to identify Testimony sentences as well as certain non-Testimony sentences with high confidence. In the second step, we employ *distant supervision*, where training data is automatically created using output of the first step. Here, we train a Bi-LSTM based sentence classifier to identify Testimony sentences.

2.1.1. Linguistic rules

Our linguistic rules are designed to ascertain that any sentence identified as Testimony sentence satisfies the following linguistic properties. Here, we use the spaCy [5] dependency parser to obtain dependency tree structure for each sentence.

- R_1: Presence of explicit (e.g., `eye-witness`, `P.W.2`) or implicit witness mentions. Implicit mentions can be pronouns (`he`, `she`), person-indicating common nouns (`landlord`, `doctor`), or actual person names (`S.I. Patil`).
- R_2: Presence of at least one statement-indicating verb like `stated`, `testified`, `narrated`.
- R_3: Within its dependency subtree, the statement verb should contain at least one of the following: a clausal complement (*ccomp*) or open clausal complement (*xcomp*).
- R_4: The statement verb should NOT have a child which negates it like `not`.
- R_5: 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 subtree (to exclude the statements by lawyers or judges).

The same set of rules are also used to identify non-Testimony sentences which are quite *similar* to Testimony sentences. Such sentences are those which satisfy the rules R_1 to R_4 but don't satisfy the rule R_5. Using the above rules, we identified 37572 Testimony sentences and 14382 non-Testimony sentences (see Table 1 for examples). In order to estimate the precision of our linguistic rules, we manually verified 200 random sentences identified as Testimony and the precision turned out to be 85%.

2.1.2. Distantly supervised Bi-LSTM based sentence classifier

As our linguistic rules are dependent on achieving correct dependency parsing, we observed that the rules fail to identify several Testimony sentences due to incorrect parsing. To overcome this, we trained a Bi-LSTM based sentence classifier which does not use

Table 1. S_1, S_2: Witness Testimony sentences identified by rules; S_3: Negative instance identified by rules for Testimony; S_4: Testimony sentence NOT identified by rules but identified by the Bi-LSTM based classifier.

S_1	`It must be noticed that P.W.-1 in his deposition stated that the appellant had taken him away in an ambassador car driven by P.W.-4 Rajib Bhuyan.`
S_2	`He further stated that the portion of the ground on which the grass was cut was shown to the Police Inspector.`
S_3	`The learned counsel stated that PWs 1, 2 and 3 must have come there to attack the appellants.`
S_4	`PW-15 further deposed that she knew Bharosa Colour Lab as she had been there several times to meet Mahesh.`

any dependency information but uses only the sequence information of the words. For training the classifier, we create the training dataset automatically by using our linguistic rules. 37572 Testimony sentences and 14382 non-Testimony sentences identified by the rules are treated as positive and negative instances, respectively. In addition, 23190 sentences are randomly selected from the rest of the corpus and treated as negative instances. Once the classifier is trained, we classify all the remaining sentences in the corpus and select 10000 sentences with highest confidence as Testimony sentences. In order to estimate the precision of our distantly supervised Bi-LSTM classifier, we manually verified 200 random sentences out of these 10000 and the precision turned out to be 75%. This classifier clearly learns more patterns over the rule based method (see Table 1). In Table 1, our rules fail to identify S_4 as a Testimony sentence because the dependency parsing fails to identify `PW-15` as the subject of the verb `deposed`. However, our Bi-LSTM based sentence classifier correctly identifies this sentences as Testimony with high confidence.

2.2. Retrieving Relevant Prior Cases

In this section, we describe two matching techniques which are used to compute *semantic* similarity between a query and a witness Testimony sentence from a prior court case.

2.2.1. Background: Semantic Roles

A syntactic or grammatical structure (such as dependency or constituency parse tree) of a sentence does not always capture full *meaning* of a sentence. E.g., consider the two sentences "`John broke the window.`" and "`The window broke.`" Here, even if the syntactic role of "`the window`" is different in both these sentences (*object* in the first sentence and *subject* in the second sentence), the underlying *semantic role* of "`the window`" is same in both of these sentences. Semantic Role Labelling (SRL) of a sentence identifies *predicate-argument* structures in the sentence such as the examples shown in Table 2. These predicate-argument structures are shown in a format adopted by PropBank [1]. In PropBank, the arguments of a predicate are numbered as A0, A1, A2 depending on the semantic role it plays. For a particular predicate, A0 is generally an *Agent* (someone who initiates the action), while A1 is a *Patient* or a *Theme* (someone who undergoes the action). No such consistent generalizations can be made across different verbs for the higher-numbered arguments. Hence, in this paper, we only consider A0 and A1 arguments of verbal predicates in witness Testimony sentences and queries. Also, as we are only focussing on predicates corresponding to event verbs, we also refer to these predicate-argument structures as *event frames*. We use MatePlus [2], a semantic role labeling (SRL) tool, for obtaining predicate-argument structures present in each sentence.

Table 2. Examples of predicate-argument structures in PropBank[1] style. The A0 (Arg0) argument plays an agent semantic role and A1 (Arg1) plays a patient/theme semantic role.

S_1: `P.W. 1 to 5 have stated that the appellant assaulted the deceased with a crow bar on his head.`
Predicate: `assaulted`, **A0 (agent)**: `the appellant`, **A1 (patient/theme)**: `the deceased`
Q_1: `Which are the cases where the appellant has attacked the deceased?`
Predicate: `attacked`, **A0 (agent)**: `the appellant`, **A1 (patient/theme)**: `the deceased`

2.2.2. Exact Semantic Match (M1)

We leverage the predicate-argument structure (as elucidated in Table 2) in a query or a sentence in a candidate prior case in retrieval. We find the similarity of a query, Q (e.g., Q_1 in Table 2) with each sentence, S (e.g., S_1 in Table 2) in a candidate prior case document **D**. To this end, we match the predicate-argument structure of Q with that of S, where the corresponding predicate and arguments are matched. That is, the *Predicate* in Q is matched with the *Predicate* in S, the *A0* in Q is matched with the *A0* in S and so on. The similarity between Q and S is defined as:

$$SIM_s(Q,S) = \frac{\sum_r match(Q_r, S_r)}{|Q|} \tag{1}$$

Here, $r \in \{Predicate, A0, A1\}$ (semantic roles), $match(.) = 1$ if there is an *exact match*, 0 otherwise. $|Q|$ is the number of *not null* arguments in the query (some of the argument values may be *null* if not detected by the SRL tool). This is done to normalise the score over the query length so that incomplete matches are penalized. In our example, the $SIM_s(Q,S)$ is $\frac{2}{3} = 0.67$. In case of a complete match (if S contained `attacked` instead of `assaulted` as *Predicate*), the value of $SIM_s(Q,S)$ will be 1. We compute the similarity at the document level, i.e. between Q and D using: $SIM(Q,D) = max_S(SIM_s(Q,S))$ (maximum of all $SIM_s(Q,S)$ values for all sentences $S \in D$).

2.2.3. Semantic Match using Event Frame Representation (M2)

Finding exact match of predicate and arguments in a query event frame (predicate-argument structure) may not be possible always due to the usage of different but semantically similar words in relevant documents. In our aforementioned example, the high semantic similarity between Q and D is not realized even though `attacked` and `assaulted` share the same semantic context. Hence, we propose to learn an embedded representation for the complete event frame structure, i.e. $\langle predicate, A0, A1 \rangle$. We train a de-noising autoencoder [6] by masking either predicate, A0 or A1 of an event frame at a time and trying to reconstruct the complete frame. We employ a simple architecture where the input layer accepts a vector (of 900 dimensions) which is a concatenation of 300-dimensional pre-trained word vectors corresponding to predicate, A0 and A1, where any one of these is masked by using a zero vector. The next layer is a fully connected dense layer of 300 dimensions. Finally, the output layer is again a 900-dimensional layer reconstructing the original concatenated vector corresponding to the complete frame. Once this autoencoder is trained, its encoder part (i.e. first two layers) is used to obtain embedded 300-dim representation of any event frame. The similarity in this case is calculated as:

$$SIM(Q,D) = max_S(cosine\_sim(Repr(Q), Repr(S))) \tag{2}$$

Here, $Repr(x)$ is the representation of a frame x. That is, we take $SIM(Q,D)$ as the maximum value of *cosine similarity* between the representations of Q and S over all the sentences $S \in D$.

Table 3. Comparative performance of all techniques. B1:BM25, B2:Doc2Vec, B3: Sentence-BERT, M1:Exact Semantic Match, M2:Semantic Match using Event Frame Representation; best values shown in bold.

Query	R-Precision (R-Prec)				
	B1	**B2**	**B3**	**M1**	**M2**
q1: Which are the cases where a husband has set his wife on fire?	0.13	0.00	0.50	0.63	0.63
q2: Which are the cases where the appellant has attacked the deceased?	0.21	0.10	0.24	0.28	0.45
q3: Which are the cases where the respondent killed the deceased?	0.00	0.00	0.0	1.00	1.00
q4: Which are the cases where the appellant demanded money?	0.06	0.13	0.0	0.56	0.75
q5: Which are the cases where the respondent has forged signatures?	0.00	0.00	0.25	0.75	0.75
q6: Which are the cases where the appellant accepted bribe?	0.00	0.00	0.17	0.33	0.50
q7: Which are the cases where an appointment was challenged?	0.14	0.14	0.00	0.43	0.57
q8: Which are the cases where an election was challenged?	0.08	0.31	0.08	0.38	0.46
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	0.06	0.07	0.22	0.64	**0.71**
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**

3. Experimental Evaluation

3.1. Dataset

Corpus: We use the Indian Supreme Court judgements from years 1952 to 2012 freely available at `http://liiofindia.org/in/cases/cen/INSC/`. There are 30,034 files containing 4,634,075 sentences and 134,329,128 tokens.
Queries: We selected 10 queries (shown in Table 3) each from different topic viz. domestic violence, homicide, forgery, corruption etc.
Ground Truth: As there is no publicly available ground truth for our queries, we use the standard *pooling technique* [7] for selection of candidate documents for annotation. We

run several ranking models (including our own techniques) and select top 20 documents for each model to form a pool which we annotate manually.

3.2. Baselines

We compare our technique with the following baselines:

1. **BM25** [8] ($B1$): A popular term scoring model based on the "bag-of-words" assumption, i.e. it *does not* consider the relative ordering of the words in the query and documents. We use the default parameter setting of the model, viz. $k_1 \in [1.2, 2]$ and $d = 0.75$.
2. **Doc2Vec** [9] ($B2$): A popular neural model that offers representation ("embeddings") of a piece of text. This is a popular neural model that offers representations ("embeddings") of a piece of text (sentence, paragraph and document). It overcomes the drawbacks of the bag-of-words models by incorporating the relative ordering of words in a text in the embeddings. We use hierarchical sampling with skip-gram model for window length=5, min-count=1.
3. **Sentence-BERT** [10] (B_3): A recent technique for obtaining sentence embeddings using Siamese-BERT networks. We used their state-of-the-art pre-trained model `bert-base-nli-stsb-mean-tokens` to obtain sentence embeddings for sentences in both query and documents. We could not fine-tune this model because of unavailability of annotated sentence pairs (with labels indicating whether they are semantically similar or not) for our experiments. Finally, the documents are ranked as per the maximum cosine similarity obtained by any of its sentence with a query sentence.

FIRE 2019 AILA track [11] contained one of the tasks which focussed on identifying relevant prior cases for a given situation. However, although the task is similar to ours, the queries are quite verbose. This is in contrast with our task where the queries are simple sentences with single verbal predicates. Hence, our techniques are not readily applicable to the task in the FIRE 2019 AILA Track. However, we use baseline techniques BM25 and Doc2Vec which are used by the most of the participating teams in this track.

3.3. Results

We evaluate the baselines and the proposed method in standard IR evaluation setup consisting of the corpus, the queries and the ground truth.

We use the following evaluation measures to evaluate the performance of our techniques as well as the baseline techniques:

1. **Average Precision (AP)**: This incorporates the relative ranking order of relevant documents; combines the joint effect of Precision and Recall.

2. **R-Precision (R-Prec)**: Precision at R, the number of relevant documents [7]

The retrieval performance of the proposed methodologies, as compared with the baselines, are shown in Table 3. Only witness Testimony sentences of the court cases in the corpus are considered for all the retrieval experiments. The proposed methods, viz. Exact Semantic Match ($M1$) and Semantic Match using Event Frame Representation

($M2$) outperform the baselines for all the queries and in both the evaluation measures, by a considerable margin. $M2$ outscores $M1$ on most queries.

To evaluate the contribution of witness Testimony sentences, we considered complete documents for BM25 as against only witness Testimony sentences. BM25 could not find even a single relevant document within top 10 for all the queries, highlighting the need for focussing only on witness Testimony sentences. Hence, we run all the experiments considering only the witness Testimony sentences. To evaluate the contribution of our distantly supervised Bi-LSTM based classifier, we applied our technique using only those Testimony sentences identified by the linguistic rules. We observed that the AP of M2 reduced from 0.73 to 0.69, stressing the importance of additional Testimony sentences identified by the distantly supervised classifier.

3.3.1. Analysis of results

Explainability of results is of paramount importance for the credibility of the system in an application area like legal domain, if the system is to be used by experts. In our proposed solution, we use semantic roles that capture an event expressed in a query. E.g., in the query q1 (`Which are the cases where a husband has set his wife on fire?`) (in Table 3), the predicate-arguments are: Predicate: `set`, A0: `husband`, A1: `wife` which semantically captures an event and matches it with a prior case where a similar event has occurred e.g., `a husband has poured kerosene on his wife and set her on fire`, based on the similarity of the semantic argument structure. We believe, this imparts more transparency and interpretability of the results in addition to the accuracy of the same. The baselines are unable to capture such nuanced semantic representations of the underlying events in a query. Our technique M2 helps in retrieving documents even if there is no exact match of the argument values in a query. E.g., for the query q2 (`Which are the cases where the appellant has attacked the deceased?`), M2 is able to retrieve the document containing the sentence `P.W. 1 to 5 have stated that the appellant assaulted the deceased with a crow bar on his head.` Although we have not used any formal notion of explainability, the proposed predicate-argument structure (semantic role) based matching schemes are able to implicitly explain the semantic similarity of a query with a prior case. However, we look to induct explainability in a more principled way in future.

It was observed that some of the queries result in much lower AP and R-Prec scores than others. This is because some queries are more general (e.g., q8) i.e., having higher number of relevant documents than some other queries which are specific (e.g., q9). The evaluation scores are dependent on the number of relevant documents retrieved at top ranks and hence are affected by this general or specific nature of the queries.

4. Conclusions and Future Work

We proposed a novel method which identifies witness Testimony sentences in Indian Supreme Court documents, extracts predicate-argument structures (or event frames) of event verbs and leverages them for prior case retrieval. The proposed method outperforms standard baselines on fine-grained queries. We look to extend our experiments on a bigger dataset with more complex queries in near future.

References

[1] Palmer M, Gildea D, Kingsbury P. The proposition bank: An annotated corpus of semantic roles. Computational linguistics. 2005;31(1):71–106.

[2] Roth M, Woodsend K. Composition of word representations improves semantic role labelling. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014. p. 407–413.

[3] Mandal A, Chaki R, Saha S, Ghosh K, Pal A, Ghosh S. Measuring Similarity Among Legal Court Case Documents. In: Proceedings of the 10th Annual ACM India Compute Conference. Compute '17. New York, NY, USA: ACM; 2017. p. 1–9. Available from: http://doi.acm.org/10.1145/3140107.3140119.

[4] McElvain G, Sanchez G, Matthews S, Teo D, Pompili F, Custis T. WestSearch Plus: A Non-factoid Question-Answering System for the Legal Domain. In: Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR'19. New York, NY, USA: ACM; 2019. p. 1361–1364. Available from: http://doi.acm.org/10.1145/3331184.3331397.

[5] Honnibal M, Montani I. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. https://spacyio/. 2017.

[6] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016.

[7] Manning C, Raghavan P, Schutze H. Introduction to information retrieval. Natural Language Engineering. 2010;16(1):100–103.

[8] Robertson SE, Walker S. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: SIGIR94. Springer; 1994. p. 232–241.

[9] Le Q, Mikolov T. Distributed representations of sentences and documents. In: International conference on machine learning; 2014. p. 1188–1196.

[10] Reimers N, Gurevych I. 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. p. 3973–3983.

[11] Bhattacharya P, Ghosh K, Ghosh S, Pal A, Mehta P, Bhattacharya A, et al. Overview of the FIRE 2019 AILA Track: Artificial Intelligence for Legal Assistance. In: FIRE (Working Notes); 2019. p. 1–12.