*Legal Knowledge and Information Systems*
*G. Sileno et al. (Eds.)*

*doi:10.3233/FAIA230990*

# LeDA: A System for Legal Data Annotation

Subinay ADHIKARY [a], Dwaipayan ROY [a], Debasis GANGULY [b], Shouvik KUMAR GUHA [c] and Kripabandhu GHOSH [a]

[a] *Indian Institute of Science Education and Research Kolkata, India*
[b] *University of Glasgow, United Kingdom*
[c] *West Bengal National University of Juridical Sciences, India*

**Abstract.** This paper presents LeDA, a system for **Le**gal **D**ata **A**nnotation. The system offers the functionality of annotating and categorising text spans representing legal concepts that capture the topic of a document, and also supports a meta-annotator to adjudicate the ground truth created by different annotators. Notably, our system supports a dynamic update of the ontology by enabling the creation of new legal concepts. Currently employed to annotate key legal concepts, LeDA aims to construct concept-based semantic representations for tasks such as similar case retrieval, and judgment prediction.

**Keywords.** Legal Data Annotation Tool, Dataset of Legal Concepts, Dynamic Ontology Update

## 1. Overview of LeDA

In legal cases, the documents often encompass lengthy and intricate sentences, making it challenging and time-consuming to thoroughly read and comprehend the entire content of a case document [1]. Therefore, extracting information from legal documents presents a formidable challenge to the research community. In response to this challenge, the research community has introduced a variety of techniques aimed at extracting information (e.g., the motive of an incident, judgment of the case, etc.) from legal documents. These techniques embrace a variety of approaches, including methods for catchphrase extraction [2], evidence identification [3] etc. Although these methods are useful for searching information from documents, none of them are capable of gaining a *thematic* or *topical* representation of the documents. The objective of our proposed annotation tool, LeDA, is to reduce the effort of annotation of legal documents with such thematic concepts that effectively capture the "aboutness" of a case document.

A typical sequence labeling annotation workflow involves selecting arbitrary spans text (e.g., entities and relations) from a document and also categorising them into a set of possible types. The main challenge in legal document annotation is that the concepts to be annotated are not as atomic as the entity names, and because of that it is rather difficult to complete the annotation process with a static set of categories for these concept types. We faced this hindrance, initially, when we started the annotation process with a standard sequence labeling tool, and it was soon realised that we need a tool that would

| Legal concept category | Description |
|---|---|
| Static Initialisation | |
| *Life_Imprisonment* | sentenced to life imprisonment |
| *Murder_on_parole* | murder during parole |
| *Second_murder* | committed second murder |
| *Physical_assault* | hurt by sharp weapon |
| *Rarest_of_the_rare_case* | the case as ”rarest of the rare case” |
| *Death_sentence* | sentenced to death |
| *Homicide_not_murder* | homicide not amounting to murder |
| *Homicide_murder* | homicide amounting to murder |
| *Political_rivalry* | incident as political rivalry |
| *Riot* | unlawful enterprise in a violent manner |
| *Juvenile_case* | considered as juvenile case |
| *Revenge* | Court identified as revenge |
| *Property_dispute* | committed as a result of property |
| *Evidence_inconsistency* | evidence of crime was not found |
| *Evidence_insufficient* | having been found inconclusive/insufficient |
| *Prosecutorial_delay_or_inability* | delayed due to prosecutorial delay |
| *Testimony_challenged* | witness testimony presented in favour of the prosecution or the defence |
| *Witness_Testimony* | witness testimony has been mentioned during the judgment |
| *Expert_witness_testimony* | includes forensic and ballistic experts |
| Dynamically added by legal experts during annotation | |
| *Prosecutorial_Delay_or_Inability* | Case is delayed due to prosecutorial delay. |
| *Investigation_agency* | This type of cases were investigated by any Central institute/state institute (e.g: CBI, NIA, ED, CID). |
| *Witness_Testimony* | Wherever witness testimony has been mentioned during the judgment and merits thereof have been discussed separately. |
| *Expert_Witness_Testimony* | This includes forensic and ballistic experts, or any other professional who is testifying about subject-matter of his expertise. |
| *Testimony_Challenged* | This will reflect whether the witness testimony presented in favour of the prosecution or the defence has been contested by the other party and also whether the court has agreed to such challenge. |

**Table 1.** A set of tags and their descriptions used in LeDA.

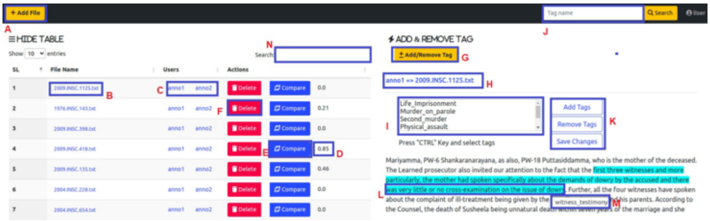
| Feature | BRAT[2] | GATE[3] | Label Studio[4] | UBIAI[5] | **LeDA** |
|---|---|---|---|---|---|
| Multiple tag | ✗ | ✗ | ✗ | ✓ | ✓ |
| Dynamic tag | ✗ | ✓ | ✓ | ✓ | ✓ |
| Adjudication | ✗ | ✗ | ✗ | ✓ | ✓ |
| Highlight | ✓ | ✓ | ✓ | ✓ | ✓ |
| IAA calculation | ✗ | ✓ | ✗ | ✗ | ✓ |
| Remote access | ✗ | ✗ | ✗ | ✓ | ✓ |
| Cost | Free | Free | Free | Proprietary | Free |

**Table 2.** Feature-wise comparison between different tools.

allow provision for the annotator to **create new concept types**, which is in fact, the key novel feature of LeDA. Table 1 reports the set of statically initialised concept types (in consultation with legal experts) along with the new tags that were created during the annotation process. Another novel feature of our tool, which is particularly important in the context of the legal concept annotation, is that of **adjudication by a meta-annotator** of multiple annotations conducted by different persons which is exactly analogous to the git-merge. We focused on independent annotation that can reduce the biases since shared documents have a chance of bias. Essentially, meta-annotators take care of conflict cases by adjudication. LeDA offers a simultaneous view of two different annotations of the same document and allows a meta-annotator to resolve the differences by choosing one or none of the conflicted entries. A comparison of LeDA with other annotation tools is presented in Table 2. Our code is made publicly available at GitHub.[1].

There are existing tools such as BRAT, GATE, DoTAT[4] etc. available for general text annotation. However, some pivotal features (i.e. Multiple tags, IAA calculation, and Remote access) necessary for annotating legal data are not available in those tools. Table 2 summarises the comparison between some of the popular annotation tools with

[1] https://github.com/subinayadhikary/LeDA



| SL | File Name | Users | Actions |
|---|---|---|---|
| 1 | 2009.INSC.1125.txt | anno1 anno2 | Delete Compare 0.0 |
| 2 | 1976.INSC.143.txt | anno1 anno2 | Delete Compare 0.21 |
| 3 | 2009.INSC.398.txt | anno1 anno2 | Delete Compare 0.0 |
| 4 | 2009.INSC.418.txt | anno1 anno2 | Delete Compare 0.85 |
| 5 | 2009.INSC.135.txt | anno1 anno2 | Delete Compare 0.46 |
| 6 | 2004.INSC.228.txt | anno1 anno2 | Delete Compare 0.0 |
| 7 | 2004.INSC.654.txt | anno1 anno2 | Delete Compare 0.0 |

anno1 => 2009.INSC.1125.txt

Life_Imprisonment<br>Murder_on_parole<br>Second_murder<br>Physical_assault

Mariyamma, PW-6 Shankaranarayana, as also, PW-18 Puttasiddamma, who is the mother of the deceased. The Learned prosecutor also invited our attention to the fact that the first three witnesses and more particularly, the mother had spoken specifically about the demands of dowry by the accused and there was very little or no cross-examination on the issue of dowry. Further, all the four witnesses have spoken about the complaint of ill-treatment being given by the his parents. According to the Counsel, the death of Susheela being unnatural death within seven years of the marriage and she

**Figure 1.** LeDA workflow. 'A': upload documents; 'B': select a document from a list; 'C' indicates that the document is annotated by both the annotators; 'D' indicates the IAA score; 'E': computes the IAA score; 'F': button to delete a document; 'G': button to add new a tag; 'H': selected document; 'I' set of tags; 'J': search documents tag-wise; 'K': buttons to add, remove or save the highlighted span and labels; 'L': highlighted span; 'M': label for highlighted span; 'N': search a document.

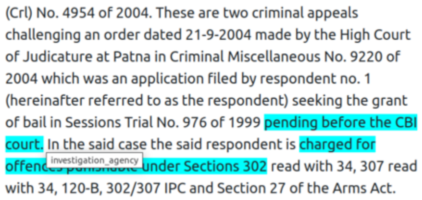
LeDA based on the available features. In the case of Doccano and YEDDA, we encountered the aforementioned issues. To study the annotation process by actual legal experts along with verifying the utility of the other features of LeDA, we have used case judgments from the Indian Supreme Court[6]. Law practitioners (from West Bengal National University of Juridical Sciences) annotated 200 legal documents using LeDA. The feedback we received on the features of LeDA was satisfactory, and most importantly, nobody suggested any new features for improvement. The rest of the paper presents more system-level details on our developed tool.

## 2. System Details

The overall system consists of a frontend and a backend. The frontend is created by using HTML, CSS, and Javascript. In the backend, we use the python-based web framework Django. For hosting our annotation tool we use PythonAnywhere[7] server. LeDA provides different interfaces for annotators and the super annotator.

**Annotator view.** Each annotator is assigned a distinct login ID and password by the administrator. These credentials are used by annotators to log in to the interface as depicted in Figure 1. Annotators select documents they are authorized to annotate. They identify granular data, assigning tags from a curated list, linked to words via “Add tags”. The process involves highlighting and tagging document details. After completion, the 'Save changes' button stores data in JSON format. To adjust annotations, the “Remove tag” function removes specific tag-word links. This cycle applies to various word sets, facilitating detailed annotation modifications. For instance, in the provided figure 1, an annotator's workflow involves selecting a document (represented as 'B'), highlighting a specific set of words (illustrated as 'L' and 'M') while associating an appropriate tag, and ultimately preserving the alterations by clicking the 'Save Changes' button (depicted as 'K') to update the JSON file. Moreover, the annotator has the capability to employ tags (referred to as 'J' in Figure 1) for searching and retrieving documents. The annotator initiates the annotation process using a predefined list of tags. If they come across any detailed information that isn't included in the current tag list, they have the option to

[6] https://indiankanoon.org/
[7] https://www.pythonanywhere.com/



(Crl) No. 4954 of 2004. These are two criminal appeals challenging an order dated 21-9-2004 made by the High Court of Judicature at Patna in Criminal Miscellaneous No. 9220 of 2004 which was an application filed by respondent no. 1 (hereinafter referred to as the respondent) seeking the grant of bail in Sessions Trial No. 976 of 1999 pending before the CBI court. In the said case the said respondent is charged for offences punishable under Sections 302 read with 34, 307 read with 34, 120-B, 302/307 IPC and Section 27 of the Arms Act.

**Figure 2.** A sample situation when a new tag , namely "`Investigation_agency`" was created during the annotation process because the highlighted text span did not thematically match with one of the statically initialised list of concept types (see Table 1).

request the super annotator to incorporate that specific fine-grained information into the existing set of tags.

**Super annotator view**. Super annotator plays a crucial role after the first phase of annotation is complete, with greater privileges than annotators. As shown in Figure 1, they upload, remove documents, and initiate annotations, `adding tags` and computing `Inter-Annotator Agreement` (IAA) [5]. We have introduced a novel approach for calculating Inter-Annotator Agreement (IAA), which significantly differs from the established method employed in GATE. As mentioned earlier the annotator can request to super annotator to add the new tag to the existing list. With the 'Add New Tag' (described in 'G') function, they enrich the tag list, in Figure 2—reflecting the 'Dynamic tag' feature, as the annotator started the annotation without a fixed ontology. To quantify the quality of annotation, computation of the Inter-Annotator Agreement (IAA) plays a crucial role, encompassing the incorporated features (as shown in 'D'). For low IAA scores (e.g., less than 0.5), they resolve the discord between annotators. Modified data is stored in JSON files via 'Save Changes'.

## 3. Conclusion and Future Work

We anticipate leveraging this meticulously annotated dataset in downstream tasks such as `prior case retrieval, judgment prediction`. As a result, LeDA can be applied to annotate various legal documents by utilizing these advanced functionalities. However, we plan to consider regular updates of the UI design incorporating new feature requests from the end users.

## References

[1] Shukla A, Bhattacharya P, Poddar S, Mukherjee R, Ghosh K, Goyal P, et al. Legal Case Document Summarization: Extractive and Abstractive Methods and their Evaluation;. .

[2] Tran V, Nguyen ML, Satoh K. Automatic catchphrase extraction from legal case documents via scoring using deep neural networks. arXiv preprint arXiv:180905219. 2018.

[3] Ghosh K, Pawar S, Palshikar G, Bhattacharyya P, Varma V. Retrieval of prior court cases using witness testimonies. In: Legal Knowledge and Information Systems. IOS Press; 2020. p. 43-51.

[4] Lin Y, Ruan T, Liang M, Cai T, Du W, Wang Y. Dotat: A domain-oriented text annotation tool. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations; 2022. p. 1-8.

[5] Bhattacharya P, Paul S, Ghosh K, Ghosh S, Wyner A. Identification of Rhetorical Roles of Sentences in Indian Legal Judgments;. .