Legal Apprentice Suite - Software Inventory

Apprentice Systems has developed a complete suite of software to support the markup and searching of documents. The primary domain is that of legal documents, but application is not limited to that domain.

The suite is built around a standard JSON document format, referred to as LSJSON (Legal Semantic JSON), which has data structures representing Case Info, Sentences (including text, classification, and global identifiers), Paragraphs, Footnotes, Notes & Tags, Groupings (designed to capture arguments and associated evidence), Decision Trees, Enrichments (based on AI) and the original source text.

The following software is involved in the creation, editing, enrichment, searching and storage of an LSJSON document. Anyone can freely use/consume that LSJSON format, but the contents of each document is protected under the author's copyright.

Legal Apprentice Software Products are stored in GitHub private repos which contributors can access by receiving and accepting an invitation from authorized ??.

The suite includes:

1. LA-MPS: Legal Apprentice (Marker, Pad, Search)

    - A browser-based client that is an integrated suite of applications for marking-up LSJSON (Marker), managing note taking across all LSJSON documents (Pad), and supporting fuzzy queries across LSJSON content (Search).

    - repo: LegalApprentice

2. LA Server -- Web Server Solution/API to support login/identification/authorization of users.
    - The server also supports realtime synchronization and message passing between the LA-MPS components through the use of web sockets.

    - repo: LoginService

3. LA TextToLSJSON -- Tools to consume legal documents in their native format (HTML, PDF, DOC, TXT) and convert them to LSJSON format.

    - This is a collection of typescript/RegEX code and libraries designed to parse documents at the sentence, word, and character levels to translate/extract the components of the LSJSON format defined above.

    - repo: LATextToLSJSON

4. LA SearchAPI -- RESTful API and web services designed to query LSJSON documents

- This software leverages elasticsearch to index and store thousands of documents by a global sentence ID, paragraph ID and document ID, in order to allow intelligent query of the content. Every aspect of the document becomes queriable, including the user's notes and tags.

- Tools to load/update LSJSON documents into elasticsearch in bulk or batch mode.

- repo: LASearchAPI

5. LA PredictAPI -- RESTful API and web services designed to support the delivery of AI models that identify and classify elements of an LSJSON document, promoting indepth understanding of the content

- In legal documents, much of the content is structured to support the components of an argument. AI models can be designed to identify cues that predict how the text fits into the legal argument being made. Sentences can be identified as to their category (e.g. Finding, Evidence, LegalRule, Citation), their polarity (for or against the plantiff) and their strength in supporting the overall argument.

- AI models can be created by many different sources, however this library creates a standard for communication and integration between the LSJSON and the rest of the Legal Apprentice Suite.

- repo LAPRedictAPI

6. LAAI Model -- Library of ongoing AI experiments/models addressing a wide range of techniques for making predictions

- The Suite contains some Python wrappers around TensorFlow to manage the training/storage/delivery of Neural Net models for tools like LAPredict

- Most of this work is documented in white papers provided to the legal community

- repos: LAAI_ModelTeach & LAAI_ModelTeach_Ruletree

Legal Apprentice Suite - Domain Specific Considerations

This suite of tools is designed to generically meet the needs of managing document and published legal cases through the ingestion, enrichment, markup, search and argument construction process. Although most documents will share these steps in common, concepts specific to the domain of reasoning and argumentation may need to be added. Special versions of LA Marker and LA TexttoLSJSON have been created where custom markup, sentence splitting and text extraction were added. Examples currently implemented include:

- Veteran PTSD Cases
- Anti-trust Cases
- New York State Appellate Cases