Semi structured data using JSON - like legal documents, missing data

By Sanchit Patil

Semi Structured Data

- Falls between the rigid structure of traditional databases and the complete lack of structure in unstructured data..
- Uses metadata, tags, or hierarchies (e.g., JSON, XML) to define elements and structure.
- Clear labels (keys) like "name," "age," and "email" offer an organized structure that’s easy to interpret and analyze.
- Common in applications like web data, IoT sensors, legal documents, and more.

Structured vs Unstructured vs Semi-structured

	Structured data	Unstructured data	Semi-structured data
Examples	• Excel spreadsheets • SQL tables	• Text documents • Images • Audio and video files	• JSON files • XML files • YAML
Use cases	• Financial records • Inventory management • Customer databases	• Social media posts • Multimedia files • Emails	• Configuration files • Log data • API responses
Data model	Relational (tables with rows and columns)	No fixed model. Data is stored as blobs, streams, or collections	Hierarchical or graph-based
Storage options	• Relational databases (e.g., MySQL, PostgreSQL) • Data warehouses	• File systems • Data lakes • Cloud storage • Document stores	• NoSQL databases (e.g., MongoDB, Couchbase) • Document stores • Key-Value stores
Analysis methods	SQL queries	• NLP • Image/video recognition • Sound/voice recognition	• Graph analysis • Path queries • Tag-based searching
Ease of search	High. Predefined schemas and indexes enable fast and accurate searches	Low. Requires advanced indexing, pattern recognition, or metadata tagging	Moderate. Flexible querying but less efficient than structured data

JSON Data

```
{
  "name": "Jane Smith",
  "email": "jane.smith@example.com",
  "preferences": {
    "newsletter": true,
    "smsNotifications": false
  }
}
```

Challenges in Handling Legal Data

- Legal documents can be highly variable with sections that may differ in structure based on jurisdiction, case type, or even specific circumstances.
- Inconsistent formatting
- Unstructured content (e.g., clauses, signatures, dates)
- Complex Relationships between different entities (for eg:- parties, dates, clauses).
- Legal documents may have missing or incomplete fields (e.g., missing signatories, dates, or clauses).

Handling Legal Data in JSON Format

- By allowing flexible key-value pairs, JSON can represent legal documents with variable structures.
- JSON allows for flexibility by allowing keys to be omitted or set as null or an empty string.
- It can be easily parsed and manipulated by programming languages.
- Can become unstructured or difficult to manage at scale without proper data validation.

Representation of NDA in JSON Format


```
{
  "ndaTitle": "Non-Disclosure Agreement",
  "parties": {
    "disclosingParty": {
      "name": "Company ABC",
      "address": "123 Business Blvd, City, State, 12345",
      "contact": "contact@companyabc.com"
    },
    "receivingParty": {
      "name": "John Doe",
      "address": null,
      "contact": "john.doe@example.com"
    }
  },
  "effectiveDate": "2025-02-01",
  "terms": {
    "confidentialInformation": "Any proprietary, technical, financial, or business information disclosed.",
    "obligations": [
      "Keep the confidential information confidential.",
      "Use the confidential information only for the purpose outlined in the agreement.",
      "Not to disclose confidential information to third parties without written consent."
    ],
    "exceptions": [
      "Information that becomes publicly available through no fault of the receiving party.",
      "Information that is already known to the receiving party at the time of disclosure."
    ]
  },
  "duration": {
    "confidentialityPeriod": "2 years",
    "agreementValidity": "5 years"
  },
  "signatures": {
    "disclosingParty": {
      "name": "Jane Smith",
      "title": "CEO, Company ABC",
      "signature": "Jane Smith",
      "dateSigned": "2025-02-01"
    },
    "receivingParty": {
      "name": "John Doe",
      "title": "Individual",
      "signature": "John Doe",
      "dateSigned": null
    }
  },
  "disputeResolution": {
    "jurisdiction": "State of California",
    "method": "Arbitration",
    "venue": null
  }
}
```


Thank
you