Natural Language Processing and Text Analytics

Group 10 Members:
Manan Khandelwal 220646
Sagar Sindhu 220372
Aakanshi Gera 220663
Lakshya Bindal 220416

1. Title of the Project

SamvidhanAI (AI-Based Legal Contract Generator)

2. Problem Statement & Objectives

Problem Statement:

The development of legal contracts presents itself as both essential and labor-intensive to organizations and professionals within the law field and to individual people. Traditional contract development heavily depends on manual drafting, thus presenting various hurdles during the process:

- ➔ Lawyers use **extensive amounts of time** during the creation process of standard contracts, along with their subsequent reviews.
- ➔ Startups, together with small businesses, encounter difficulties in **paying for legal professionals** to handle standard contracts that they need to sign.
- ➔ Manual entry creates **multiple problems** by causing human mistakes that produce errors and omitting essential clauses as well and creating document mismatches.
- ➔ Standardized compliance and consistency challenges persist among firms that must process numerous contracts daily.

Real-World Example Scenarios:

1. **Start-ups** with contractors and **freelance** personnel find it unsustainable to hire lawyers to draft the necessary Non-Disclosure Agreements, along with employment documents and partnership contracts, because it results in high expenses.
2. The **Real Estate Industry** requires proper lease agreement creation, which contains exact tenant information along with payment protocols and legal requirements of the jurisdiction.
3. **Subscription-based services** operated by E-commerce companies need contracts that fulfill legal requirements for consistency.

Novelty in Our Project:

1. **Rule-Based NLP for Contract Generation:** The process takes entity extraction differs from the usual template-based methods by both taking suitable entities and ensuring their correct position inside predefined templates.

2. **Automation Without LLMs:** This system stands out with its inexpensive design and understandable structure while operating at low costs.

3. Step-by-Step Building Methodology

Step 1: Data Collection & Preprocessing

➔ **Dataset:** CaseHOLD (Case Holdings On Legal Decisions)

CaseHOLD: The dataset contains 53,000+ multiple-choice tests that evaluate systems in their ability to select the correct legal ruling from a given case. Users encounter legal phrases that contain one correct answer and four incorrect options in each examination. The dataset trains AI models to carry out legal thinking and analyze case law content.

➔ **Data Cleaning and Preprocessing:**
- ◆ Tokenization
- ◆ Remove stop words and legal jargon.
- ◆ Normalize text (lowercasing, punctuation removal, etc.).

Step 3:Applying TF-IDF and GloVe. 6 B. 100 model

➔ **TF-IDF (Term Frequency-Inverse Document Frequency):**
The technique uses word frequency and document comparison with the entire text collection to produce numerical scores that represent word importance. The method enables users to detect fundamental legal terminology that includes "confidentiality" alongside "liability".

➔ **GloVe (Global Vectors for Word Representation):**
The model uses GloVe 6 B. 100d embeddings, which provide dense vectors that represent semantic word meanings. GloVe operates differently from TF-IDF because it considers word context. For example, it distinguishes the word meaning between "employer" and "employee".

➔ **Why both?**
TF-IDF gives term relevance, while GloVe captures contextual similarity. Together, they provide an understanding of contract types and help with classification.

Step 2: Named Entity Recognition (NER) for Clause Extraction

- **Tools Used:** Spacy, NLTK
- **Entities to Extract:**
    - ➔ **PARTIES:** Names of individuals or companies
    - ➔ **DATES:** Contract start and end dates
    - ➔ **DURATION:** Contract validity
    - ➔ **PAYMENT TERMS:** Financial obligations
    - ➔ **GOVERNING LAW:** Applicable jurisdiction

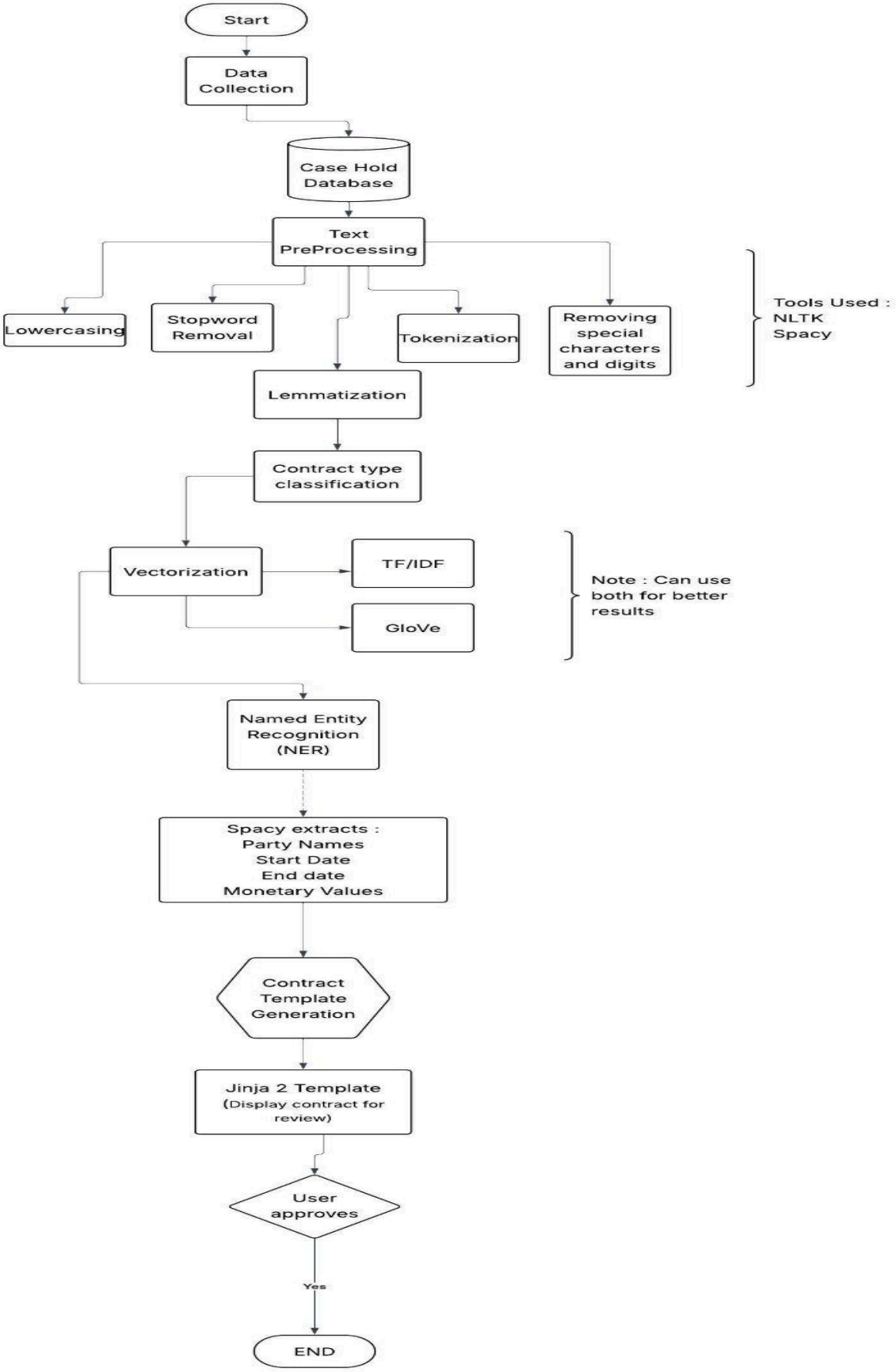
Step 3: Contract Template Generation

- **Technology Used:** Jinja2 Template (A predefined format for legal documents)
- **Process:**
    - ➔ Create predefined templates with placeholders.
    - ➔ Dynamically fill placeholders with extracted data.

Step 4: Model Evaluation & Improvement

- **Performance Metrics:**
    - ➔ Precision & Recall (for NER accuracy)
    - ➔ Execution Time (contract generation speed)
    - ➔ User Feedback & Iteration

4. System Diagrams



5. Architecture Design