Scope Document for Legal Document Simplification System (LDSS)

Project Overview

The **Legal Document Simplification System (LDSS)** aims to transform complex legal documents into simplified, easy-to-understand content. It is designed to assist non-lawyers by demystifying legal jargon while retaining the original intent and legal meaning. The system will provide multi-format input support (text, PDF, and image), offer customization options for simplicity levels, include text-to-speech capabilities, and support local languages, such as Konkani.

Objectives

1. **Simplify Complex Legal Content**: Convert intricate legal terms, clauses, and phrases into plain, understandable language for diverse user groups.
2. **Enhance Accessibility**: Enable users to upload documents in various formats, including text, PDF, and images, for analysis and simplification.
3. **Incorporate Local Language Support**: Provide simplification in regional languages, starting with Konkani, to cater to local communities.
4. **Customizable Simplicity Levels**: Allow users to adjust the complexity of the output based on their education and understanding.
5. **Text-to-Speech Functionality**: Make legal content accessible to visually impaired users and those who prefer audio.

Requirements

Functional Requirements

1. **Input Processing**:
   - Accept inputs in text, PDF, and image formats.
   - Extract text from images using Optical Character Recognition (OCR).
2. **Text Simplification**:
   - Identify legal terms and clauses.
   - Simplify text based on selected simplicity levels.
3. **Language Support**:
   - Simplify content in English and Konkani, with scope for additional languages.
4. **Customizable Simplicity Levels**:
   - Allow users to select simplicity levels (basic, intermediate, advanced).
5. **Text-to-Speech Conversion**:
   - Generate audio outputs of simplified content.
6. **Output Presentation**:
   - Display original and simplified texts side by side.

    - Provide download options for simplified text and audio outputs.

Non-Functional Requirements

1. **Accuracy**:
    - Ensure high fidelity between simplified text and the original legal meaning.
2. **Performance**:
    - Process documents quickly, including large and complex files.
3. **Scalability**:
    - Handle multiple users and concurrent document processing.
4. **Accessibility**:
    - Design an intuitive interface for users with varying technical expertise.

Execution Process

Phase 1: Research and Data Collection

1. Gather datasets of legal documents, glossaries, and plain-language examples.
2. Collect local language data (e.g., Konkani legal texts or translations).

Phase 2: Technology Selection and Setup

1. Select NLP libraries and models:
    - Hugging Face Transformers (e.g., LegalBERT, T5).
    - SpaCy for Named Entity Recognition (NER).
2. Choose OCR tools for image-to-text conversion:
    - Tesseract OCR or Google Vision API.
3. Use gTTS (Google Text-to-Speech), Pyttsx3 or AWS Polly for audio output.

Phase 3: Core Development

1. **Input Module**:
    - Implement text, PDF, and image upload functionality.
    - Integrate OCR for text extraction from images.
2. **Simplification Engine**:
    - Develop rule-based and ML-driven systems for text simplification.
    - Fine-tune pre-trained models for domain-specific tasks.
3. **Language Module**:
    - Konkani translation and simplification.
    - Include multilingual support for future expansions.
4. **Customization Module**:
    - Add simplicity-level options and adjust outputs.
5. **Text-to-Speech Module**:
    - Convert simplified text into audio format.

Phase 4: Frontend Development

1. Design a user-friendly interface:
    - Upload section for documents/images.
    - Dual-view of original and simplified text.
    - Simplicity level sliders and audio playback controls.
2. Implement responsive design for desktop and mobile users.

Phase 5: Testing and Validation

1. Validate text extraction from images.
2. Test text simplification accuracy and fidelity.
3. Ensure smooth functioning of multilingual and text-to-speech features.
4. Collect feedback from legal experts and non-lawyer users.

Phase 6: Deployment

1. Containerize the application using Docker.
2. Deploy on a cloud platform (AWS, Azure, or GCP).
3. Ensure scalability with load balancers and caching mechanisms.

Additions to Core Features

Local Language Support

- Translate legal content into Konkani.
- Train models on regional legal datasets for accurate contextual translation.

Text-to-Speech

- Convert simplified content into audio files.
- Provide playback and download options for generated audio.

Simplicity Levels

- Implement user-adjustable sliders to select complexity levels.
- fine-tune simplification based on user preferences.

Image Input Support

- Use OCR to extract text from scanned legal documents or images.
- Validate extracted text before processing it for simplification.

Tech Stack

Frontend

- **Framework**: React.js, Streamlit

- **Styling**: Material-UI or TailwindCSS

Backend

- **Framework**: FastAPI
- **Libraries**: Hugging Face Transformers, SpaCy, Tesseract OCR

Database

- **Primary Storage**: PostgreSQL or MySQL for storing metadata.
- **Cache**: Redis for improving performance.

Text-to-Speech

- **Library**: gTTS, Pyttsx3, AWS Polly

Deployment

- **Containerization**: Docker
- **Cloud Hosting**: AWS, Azure, or GCP

Challenges and Mitigation

Challenges

1. Ensuring legal accuracy during simplification.
2. Handling diverse document formats and structures.
3. Supporting regional language nuances (e.g., Konkani grammar and vocabulary).
4. Balancing performance with advanced features like OCR and TTS.

Mitigation Strategies

1. Validate outputs with legal experts to ensure accuracy.
2. Use pre-processing pipelines to standardize document formats.
3. Collaborate with linguistic experts for local language support.
4. Optimize code and deploy scalable infrastructure to maintain performance.

Project Duration

The estimated duration for the completion of the Legal Document Simplification System (LDSS) is **2 months**. This timeline includes the research, development, testing, and deployment phases.

Scope of our work

The team's **focus** will be on developing the **simplification process**, ensuring that complex legal documents are accurately converted into plain language.

Not in the Scope of our work

1. Implementing **customisable simplicity levels** to allow users to adjust the complexity of the output.

2. Additionally, Integrating **local language support**, such as Konkani, to make the system accessible to regional users.

Conclusion

The **Legal Document Simplification System (LDSS)** is a comprehensive solution designed to bridge the gap between legal complexity and public accessibility. By incorporating features like multilingual support, text-to-speech, simplicity customization, and image processing, LDSS aims to make legal content more inclusive and user-friendly. This project has the potential to significantly benefit individuals, legal professionals, and businesses by simplifying legal documentation and fostering better understanding.