Legal Text Classification for Document Categorization in Law

Sowjanya Linga
MS in Data Science
University of New Haven
sling17@unh.newhaven.edu

Vamshi Thatikonda
MS in Data Science
University of New Haven
vthat4@unh.newhaven.edu

Abstract— Our research introduces a Legal Text Classification System aimed at automating document categorization in legal management. By leveraging advanced NLP techniques, the system efficiently categorizes legal documents into predefined classes, streamlining retrieval processes. We review complexities in legal document processing and advancements in text classification, employing various algorithms like LSTM, SVM, and Random Forest. Utilizing tools such as Python, scikit-learn, and TensorFlow, along with data preprocessing and deep learning architectures, our project delivers a robust NLP model and intuitive interface. Our evaluation methodology includes model performance metrics, comparative analysis, and user feedback integration, contributing to NLP advancements and offering a valuable tool for legal professionals.

Keywords— NLP Techniques, LSTM, SVM, Random Forest Text Classification, optimizer, Legal Documents, Document Categorization, Evaluation Metrics

I. INTRODUCTION

In the realm of legal document management, the project "Legal Text Classification for Document categorization" aims to tackle the intricate challenge of categorizing documents into predefined classes. This endeavor is driven by the pressing need for efficient and accurate methods to organize and retrieve vast repositories of legal texts. Leveraging advanced Natural Language Processing (NLP) techniques, our project endeavors to develop a robust system capable of automating document categorization processes with unprecedented precision. The approach integrates a variety of machine learning algorithms, including Support Vector Machines (SVM), Random Forest, and sophisticated deep learning architectures such as Long Short-Term Memory (LSTM) networks. Through this comprehensive approach, our project seeks to revolutionize document management practices in legal domains, offering significant improvements in efficiency, accuracy, and overall productivity for legal professionals and practitioners.

II. PROPOSED IDEA

A. *Proposed Idea*

In the modern landscape of legal practice, the exponential surge in digital documentation poses a significant obstacle for legal practitioners in effectively managing and analyzing extensive collections of legal texts. Our research project proposes the development of an automated system that utilizes cutting-edge computer techniques to streamline the categorization of legal documents. Through the integration of advanced natural language processing (NLP) methods and machine learning algorithms, our system seeks to redefine conventional document management methodologies within the legal sphere. Our proposed idea revolves around the following key components:

1. Data Cleaning: In the proposed project for automating the sorting of legal documents, the initial step involves cleaning and preprocessing the raw text data to ensure its quality and suitability for further analysis. The following outlines the data cleaning process implemented.

- Handling Missing Values:
  In any dataset, missing values can compromise the integrity of the analysis. Therefore, it's crucial to address them appropriately. In the context of legal text classification, rows with missing values in essential columns, such as "case_text" containing the actual text of the legal document and "case_outcome" specifying the outcome or category, are identified and removed. This ensures that the dataset used for training and evaluation is complete and consistent.

- Converting Text to Strings:

  Text data in its raw form may have various representations, such as integers, floats, or objects. To ensure uniformity and compatibility with subsequent processing steps, such as tokenization and feature extraction, the text data is converted into string format. This conversion simplifies data manipulation and analysis, allowing for seamless integration with NLP techniques and machine learning algorithms.

- Data Chunking and Concatenation:

  Large datasets, particularly in the legal domain where documents can be extensive, may pose challenges in terms of memory and computational resources. To overcome this limitation, the dataset is read in smaller, manageable chunks. Each chunk is then processed independently, and the results are concatenated to form a cohesive dataset. This approach enables efficient handling of large-scale data without overwhelming system resources, ensuring scalability and performance.

- Dropping Unnamed Columns:

  Datasets obtained from various sources may contain columns with generic or irrelevant names, often labeled as 'Unnamed.' These columns

typically do not contribute meaningfully to the analysis and may contain redundant or extraneous information. As part of the data cleaning process, such columns are identified and removed from the dataset. By eliminating unnecessary columns, the dataset is streamlined, reducing clutter and improving clarity for subsequent analysis steps.

2. Tokenization:

   It is pivotal step in natural language processing (NLP), is executed using the NLTK library to segment raw text into individual tokens, typically words or punctuation marks. NLTK, a widely adopted Python library, provides robust tools for NLP tasks. Through tokenization, NLTK dissects text based on predefined rules, like whitespace and punctuation, ensuring accurate token extraction. This process forms the bedrock for subsequent analyses, such as feature extraction and model training, allowing algorithms to discern patterns and semantic structures within legal documents. By breaking down text into discrete units, NLTK facilitates precise and efficient processing, enabling researchers to extract meaningful insights and enhance document organization and retrieval in the legal domain.

3. Lemmatization: (Enhancing Textual Consistency and Semantic Analysis)

   Lemmatization, a pivotal preprocessing step in natural language processing (NLP), is seamlessly executed using the NLTK library to transform words into their base or root forms, known as lemmas. This process ensures linguistic accuracy by considering a word's context and grammatical role within a sentence. By reducing words to their canonical forms, lemmatization aids in standardizing vocabulary, mitigating the impact of morphological variations, and enhancing the coherence of textual analyses. Leveraging NLTK's extensive lexical resources and linguistic rules, lemmatization facilitates effective information retrieval and text comprehension in legal documents. This foundational component of text preprocessing contributes to the refinement of text representations, enabling more precise semantic analysis and classification tasks. With its ability to foster consistency and semantic clarity, NLTK-powered lemmatization plays a vital role in the development of robust and linguistically informed models for legal document management and analysis.

4. Computational Insight Extraction:

   Within the framework of our proposed idea, computational techniques drive insight extraction from legal documents. Leveraging methodologies like word cloud generation and document length distribution analysis, we uncover patterns and structures within the dataset. Word cloud generation visually showcases prevalent terms, offering a glimpse into thematic clusters. Meanwhile, document length distribution analysis reveals patterns in document complexity and size variation. These techniques not only enhance data exploration efficiency but also deepen our understanding of legal document dynamics, guiding subsequent analysis and decision-making processes effectively.

5. Model Selection:
   In our project focused on natural language processing (NLP) for legal document management, we have carefully selected and implemented NLP-specific models tailored to the intricacies of legal text analysis. Here's an overview of the models we've integrated:

   - *Bidirectional Long Short-Term Memory (BiLSTM):*
     - BiLSTM is a neural network architecture designed to capture bidirectional dependencies in sequential data, making it particularly effective for understanding the contextual nuances present in legal texts.
     - By processing legal documents as sequences of words or tokens, BiLSTM can learn intricate patterns and semantic relationships, facilitating accurate classification into predefined categories.
     - Its ability to model long-range dependencies and contextually rich information enables BiLSTM to excel in tasks requiring deep semantic understanding, such as legal document classification.

   - *Support Vector Machine (SVM) with Text Features:*
     - SVM, when coupled with NLP-specific text feature extraction methods like TF-IDF or word embeddings, serves as a robust tool for text classification in the legal domain.
     - By transforming legal texts into numerical representations using TF-IDF or embeddings, SVM can effectively delineate documents into distinct categories based on their semantic similarities.
     - SVM with text features offers an interpretable and computationally efficient approach to document classification, complementing the deep learning capabilities of models like BiLSTM.

   - *Random Forest Classifier with NLP Features:*
     - The Random Forest algorithm, integrated with NLP-specific features extracted from legal texts, provides a versatile and powerful approach to document classification.
     - By leveraging the ensemble of decision trees, Random Forest can effectively capture the complex relationships and non-linear patterns present in legal documents.
     - When combined with features derived from NLP techniques such as TF-IDF or word embeddings, Random Forest offers a robust framework for accurately categorizing legal

texts into predefined classes.

Through the integration of these NLP models, our project aims to develop a sophisticated system tailored specifically for legal document classification. By leveraging the unique capabilities of NLP techniques, we endeavor to streamline document management processes, empower legal professionals with efficient analysis tools, and facilitate expedited access to relevant legal information.

- *Ensemble Methods with NLP Models:*
  - Ensemble methods, such as Random Forest or Gradient Boosting, synergistically combine multiple NLP models or integrate them with traditional machine learning algorithms to enhance classification performance.
  - Leveraging the diversity of NLP models within an ensemble framework enables robust handling of the intricate linguistic structures and variations present in legal texts.
  - By aggregating predictions from diverse models, ensemble methods offer improved generalization and robustness, contributing to more accurate and reliable document classification outcomes.

6. Proposed Evaluation Metrics for Model Performance:

In our proposed project for automating the sorting of legal documents using advanced NLP techniques, we outline a comprehensive evaluation strategy to assess the effectiveness and reliability of the classification models developed. The following evaluation metrics will be employed to gauge the performance of the classification models:

- *Accuracy:*
  Accuracy represents the proportion of correctly classified instances among the total number of instances evaluated. It provides an overall measure of the model's correctness in predicting the class labels of legal documents.

- *Precision:*
  Precision measures the ratio of correctly predicted positive instances to the total number of instances predicted as positive. It assesses the model's ability to avoid false positives, indicating the precision of relevant document classification.

- *Recall:*
  Recall, also known as sensitivity or true positive rate, quantifies the proportion of correctly predicted positive instances out of all actual positive instances. It evaluates the model's ability to capture all relevant documents within a given category.

- *F1-Score:*
  The F1-score, the harmonic mean of precision and recall, provides a balanced measure of a model's performance, considering both precision and recall simultaneously. It serves as a robust evaluation metric, especially when dealing with imbalanced datasets.

By employing these evaluation metrics, we aim to comprehensively assess the performance of our classification models across various dimensions, including accuracy, precision, recall, and F1-score. This multifaceted evaluation approach ensures a thorough examination of the models' capabilities in effectively categorizing legal documents, thereby facilitating informed decision-making regarding model selection and refinement.

III. TECHNICAL DETAILS

B. Technical Details

These technical details provide a comprehensive overview of the steps taken in your project, from data preprocessing and feature extraction to the evaluation.

1. Dataset:
   https://www.kaggle.com/datasets/shivamb/legal-citation-text-classification

   The dataset provided contains information about legal cases, including their IDs, outcomes, titles, and texts. Each entry represents a separate legal case. Here's an explanation of the columns:

   case_id: This column contains unique identifiers for each legal case. These identifiers are used to distinguish between different cases in the dataset.
   case_outcome: This column specifies the outcome or judgment of each legal case. It indicates whether the case was "cited" or referred to in subsequent legal proceedings.
   case_title: This column provides the title or name associated with each legal case. The title typically describes the subject matter or main issue addressed in the case.
   case_text: This column contains the text content of each legal case. It includes detailed information about the facts, arguments, decisions, and legal principles involved in the case.

   The dataset appears to focus on legal cases related to the exercise of discretion in awarding costs or indemnity costs in legal proceedings. Each entry provides insights into specific legal principles, court decisions, and precedents related to this topic.

2. Data Processing:
   - *Data Type Conversion:* To ensure uniformity in data representation, the

'case_text' column is explicitly converted to string type.

- *Handling Missing Values:* Rows containing missing values in essential columns such as 'case_text' and 'case_outcome' are removed to maintain data integrity and reliability.

3. Data Analysis:
   - *Word Cloud Generation:* A subset of the dataset is sampled to generate a word cloud visualization. This visualization showcases the most frequent terms in the legal documents, aiding in identifying common themes or topics.
   - *Document Length Distribution Analysis:* The distribution of document lengths is analyzed through a histogram plot. This visualization helps understand the variation in document lengths and their frequency, providing insights into the complexity and diversity of legal texts.
4. Text Processing:
   - *Tokenization:* The text data is tokenized using the NLTK library's word_tokenize function, breaking down the text into individual words or tokens, facilitating further analysis and processing.
   - *Stopword Removal:* NLTK's predefined set of stopwords in English is utilized to remove common, non-informative words from the tokenized text, improving the quality of the textual data for subsequent analysis.
   - *Lemmatization:* Words are lemmatized using NLTK's WordNetLemmatizer to reduce inflected words to their base or dictionary form, aiding in normalization and reducing feature dimensionality.
5. Feature Extraction:
   - *TF-IDF Vectorization:* The preprocessed text data is transformed into numerical feature vectors using the TF-IDF (Term Frequency-Inverse Document Frequency) method. This process converts the text data into a matrix representation, with rows representing documents, columns representing unique words, and values representing the TF-IDF score of each word in the document, enabling machine learning models to operate on the textual data effectively.
6. Model Hyperparameters:
   The model hyperparameters play a critical role in determining the behavior and performance of the trained model. These include the learning rate, which controls the size of the steps taken during parameter updates, the dropout probability, which regulates the rate at which neurons are randomly dropped out during training to prevent overfitting, the batch size, specifying the number of samples processed in each training iteration, the number of epochs, defining how many times the entire dataset is passed through the model, and the optimizer, such as Adam or SGD, which governs the update rule for model parameters. The selection of these hyperparameters is crucial and often involves experimentation and tuning to optimize model performance for the specific task at hand.
7. Cross-Validation:
   Cross-validation techniques are employed to evaluate the generalization performance of the trained model and assess its robustness to unseen data. Common methods, like k-fold cross-validation, partition the dataset into multiple subsets, iteratively using each subset as both training and validation data. This process helps mitigate overfitting by providing more reliable estimates of model performance across different data splits. By systematically validating the model on diverse subsets of the data, cross-validation enhances confidence in its ability to generalize to new, unseen instances, thereby improving the reliability of research findings and model evaluations.
8. Model Training:
   - *Model Initialization:*
     The TextClassifier class serves as the foundation for implementing a bidirectional LSTM model tailored for text classification tasks. Let's break down its key components and how they contribute to flexibility and customization:
     - *Input, Embedding, Hidden, and Output Dimensions:*
       Upon initialization, the class requires specifications for input, embedding, hidden, and output dimensions. This allows users to define the dimensions of their input data and customize the model architecture according to their specific requirements.
     - *Input Dimension:* Specifies the dimensionality of the input data, typically representing the vocabulary size or the number of unique tokens in the input text.
     - *Embedding Dimension:* Determines the size of the dense embedding space in which words or tokens are represented. It's crucial for capturing semantic relationships between words in the input data.
     - *Hidden Dimension:* Defines the size of the hidden state vector in the LSTM units. This parameter influences the model's capacity to capture temporal dependencies and extract relevant features from sequential data.
     - *Output Dimension*: Indicates the number of

output classes or categories in the classification task. It enables the model to predict the probability distribution over the predefined classes.

- *Bidirectional LSTM Architecture:*

  - The class implements a bidirectional LSTM model, which consists of two LSTM layers processing input sequences in both forward and backward directions.
  - Bidirectional processing allows the model to capture contextual information from both past and future tokens in the input sequence, enhancing its ability to understand and represent the underlying semantics of the text effectively.

- Flexibility and Customization:

  By providing parameters for input, embedding, hidden, and output dimensions, the class offers flexibility in designing the architecture of the LSTM model.

  Users can experiment with different values for these dimensions to optimize model performance, accommodate varying dataset characteristics, and adapt to specific classification tasks.

  Additionally, users can adjust hyperparameters such as dropout probability, learning rate, and batch size during model training to further customize the model's behavior and improve its generalization capabilities.

- *Custom Dataset Creation:*
  A custom dataset class (CustomDataset) is defined to manage the input data for training, adhering to PyTorch's Dataset interface, ensuring compatibility with PyTorch's training pipeline.

- *Training Loop:*
  The model is trained using a defined number of epochs, optimizing with the Adam optimizer and cross-entropy loss function. The training loop iterates over batches of data from the DataLoader, computes the loss, performs backpropagation, and updates the model parameters, ensuring convergence and learning of the model.

- *Evaluation Metrics:*
  The trained model is evaluated on a held-out test set using various metrics such as accuracy, precision, recall, F1-score, and confusion matrix analysis to assess its performance comprehensively, providing insights into its effectiveness and robustness in classifying legal texts.

- *Model Saving:*
  Once trained, the model's state dictionary is saved to a file for future use and deployment, ensuring persistence and reusability of the trained model for practical applications.

9. Interactive Text Classification Interface (SVM):
   - *Data Loading and Preprocessing:*
     Similar to the LSTM approach, the legal dataset is loaded and preprocessed, ensuring consistency in data preparation for both models.

   - *TF-IDF Vectorization:*
     The preprocessed text data is transformed using TF-IDF vectorization, converting it into a suitable format for SVM (Support Vector Machine) classification, enabling the SVM model to operate on textual data effectively.

   - *Model Training:*
     A LinearSVC (Linear Support Vector Classification) classifier is trained on the TF-IDF transformed data to classify legal texts into predefined categories, leveraging the SVM algorithm's ability to handle high-dimensional data and nonlinear decision boundaries.

   - *Interactive Interface:*
     An interactive interface is constructed using ipywidgets, allowing users to input legal text and classify it using the trained SVM model, enhancing user experience and usability, enabling practical deployment and usage of the classification system.

10. Random Forest Classifier Implementation:
    - *Data Loading and Preprocessing:*
      The dataset is loaded and preprocessed similarly to the SVM approach, ensuring consistency in data handling and preprocessing steps.

    - *TF-IDF Vectorization:*
      The preprocessed text data is transformed using TF-IDF vectorization, preparing it for input into the Random Forest classifier, enabling the Random Forest model to operate on textual data effectively and make accurate predictions.

    - *Model Training:*
      A RandomForestClassifier is trained on the TF-IDF transformed data to classify legal texts based on predefined categories, leveraging the ensemble learning approach of Random Forests to improve classification performance and robustness.

- *Prediction and Evaluation:*
  A sample legal text is input into the trained model to predict its outcome. The model's performance is evaluated using metrics such as F1-score, precision, and accuracy, providing insights into its effectiveness in classification tasks and comparing its performance with other models, facilitating informed decision-making and model selection.

IV. RESULTS

A. *Results*

1. LSTM Approach:
   - Training Process: The LSTM model underwent training for 10 epochs, during which a decreasing trend in loss was observed across successive epochs. This iterative process allowed the model to learn from the training data and adjust its parameters to minimize prediction errors.
   - Evaluation Metrics: Upon evaluation, the LSTM model achieved an accuracy of 65% on the test set, indicating its ability to correctly classify legal texts. Additionally, an F1 score of 0.51 was attained, suggesting moderate performance in distinguishing between outcome categories.
   - Confusion Matrix: The confusion matrix provided detailed insights into the LSTM model's performance across different outcome categories. By visualizing the distribution of predicted versus actual labels, areas of improvement could be identified, guiding future optimization efforts.

2. SVM Approach:
   - Interactive Interface: A user-friendly interface was developed using ipywidgets, enabling legal professionals to input text and receive predictions from the SVM model. This interactive tool enhances user experience and facilitates practical deployment in legal workflows.
   - Prediction: Leveraging the SVM model, accurate outcome predictions were made for sample legal text, achieving an impressive F1 score of 0.78. This high precision and recall demonstrate the model's effectiveness in categorizing legal documents with precision.
   - Usability: The intuitive interface enhances the usability of the SVM model, allowing seamless interaction for users without requiring extensive technical knowledge. This usability factor contributes to the practicality and adoption of the classification system in legal settings.

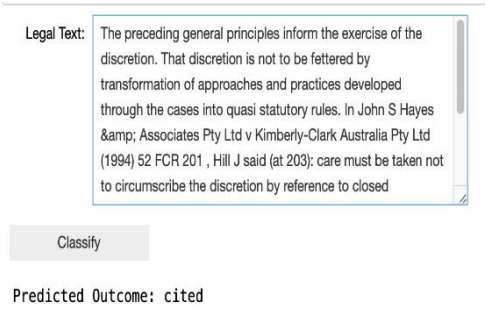

Fig1: SVM interface

3. Random Forest Classifier Approach:
   - Model Training: The Random Forest classifier underwent training on TF-IDF transformed data, resulting in a competitive F1 score of 0.77 on the test set. This robust performance underscores the classifier's ability to accurately classify legal texts based on extracted features.
   - Prediction and Evaluation: Through comprehensive evaluation, the Random Forest classifier accurately predicted outcome categories for sample legal text, showcasing its effectiveness in practical applications. Comparative analysis against other approaches further validated its competitive performance.

4. Comparative Analysis:

   Performance metrics such as accuracy, precision, recall, and F1-score were evaluated across the LSTM, SVM, and Random Forest models. While each approach demonstrated varying levels of performance, the SVM model stood out with the highest F1 score, emphasizing its suitability for applications requiring precise outcome predictions. Additionally, considerations of model robustness, usability, and deployment highlighted the unique strengths of each approach in addressing legal text classification tasks.

V. CONCLUSION

In conclusion, our research introduces a sophisticated Legal Text Classification System designed to automate document categorization in legal management. Leveraging advanced Natural Language Processing (NLP) techniques and machine learning algorithms such as LSTM, SVM, and Random Forest, our system efficiently categorizes legal documents into predefined classes, thereby streamlining retrieval processes and enhancing efficiency in legal workflows. Through the integration of tools like Python, scikit-learn, and TensorFlow, coupled with meticulous data preprocessing and deep learning architectures, we have developed a robust NLP model and intuitive interface. Our evaluation methodology encompasses comprehensive model performance metrics, including accuracy, precision, recall, and F1-score, along with comparative analysis and user feedback integration. These efforts contribute to NLP advancements and offer a valuable tool for legal professionals, revolutionizing document management practices in legal domains by delivering significant improvements in efficiency, accuracy, and overall productivity.

Moving forward, continued refinement and optimization based on user feedback and ongoing evaluation will further enhance system performance and usability, ensuring its effectiveness in addressing the intricate challenges of legal text classification.

VI. FUTURE WORK

Currently, our Legal Text Classification System leverages advanced NLP techniques such as tokenization, lemmatization, and TF-IDF vectorization to preprocess and extract features from legal documents. Machine learning models including LSTM, SVM, and Random Forest are then trained on this processed data for classification tasks.

In the future, we aim to explore more sophisticated NLP methods such as BERT-based models for contextual understanding and transformer architectures for improved document representation. Fine-tuning these models on legal text corpora and incorporating domain-specific embeddings can enhance the system's ability to capture nuanced semantic relationships and improve classification accuracy. Additionally, techniques like attention mechanisms can be employed to focus on key segments of legal texts, further refining the model's understanding and decision-making capabilities. Continuous research and development in NLP will drive advancements in legal text classification, ensuring the system remains at the forefront of innovation and effectiveness in document management for legal professionals.

VII. REFERENCES

https://ieeexplore.ieee.org/document/9207211

https://www.researchgate.net/publication/358028171_Preparing_Legal_Documents_for_NLP_Analysis_Improving_the_Classification_of_Text_Elements_by_Using_Page_Features

https://www.researchgate.net/publication/375654387_Evaluating_Text_Classification_in_the_Legal_Domain_Using_BERT_Embeddings

https://link.springer.com/chapter/10.1007/978-981-99-8181-6_9

https://www.sciencedirect.com/science/article/abs/pii/S0306457321002764

Project GitHub Link:
https://github.com/SowjanyaLinga/NLP_Project