# Legal Domain Similarity Analysis

Jiya Borikar, Lohitha Kanisettypalli, Harsda Shrivastava, Peeta Basa Pati

*Department of Computer Science and Engineering*

*Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India*  bl.en.u4aie23010@bl.students.amrita.edu, bl.en.u4aie23061@bl.students.amrita.edu, bl.en.u4aie23047@bl.students.amrita.edu, bp\_peeta@blr.amrita.edu

***Abstract* — This project focuses on developing a system to measure the similarity between legal texts using NLP and machine learning techniques. Various approaches, including TF-IDF, cosine similarity, word embeddings, and sentence transformers, are applied to capture both lexical and semantic relationships. Preprocessing steps such as missing value handling and normalization improve the accuracy of similarity detection. Feature representation plays a crucial role in enhancing classification performance, and different machine learning techniques are evaluated for their effectiveness in legal text analysis. The system identifies and groups similar legal documents, aiding case law research, contract comparison, and precedent analysis. By moving beyond traditional keyword-based methods, it improves efficiency and accuracy in legal text retrieval.**

***Keywords— Legal Document Analysis, Text Similarity, NLP, Machine Learning, Classification ,KNN, Confusion matrix***

I. INTRODUCTION

Legal texts, including court rulings, contracts, and statutes, contain complex structures and domain-specific terminology that make textual analysis a challenging yet crucial task. The ability to measure the similarity between legal documents is essential for various applications, such as case law research, precedent analysis, contract comparison, and legal document classification. Traditional keyword-based search methods often fail to capture the nuanced relationships between legal texts, necessitating the use of more advanced techniques.

This project, Legal Domain Similarity Analysis, aims to develop a system that evaluates the degree of similarity between legal texts and outputs a similarity score. Given an input document, the system will compare it against a set of legal texts using natural language processing (NLP) and machine learning techniques. The project will explore different similarity measurement approaches, including lexical, syntactic, and semantic similarity techniques. Potential methods include TF-IDF, cosine similarity, word embeddings (Word2Vec, GloVe, BERT), and sentence transformers.

A robust similarity detection system has significant implications in the legal field. It can assist legal professionals in retrieving relevant cases efficiently, identifying related legal arguments, detecting contract similarities, and improving automated legal research tools. By leveraging modern NLP approaches, this project aims to enhance the accuracy and efficiency of legal text comparison, reducing the time and effort required for manual legal document analysis.

II. LITERATURE SURVEY

The paper "Legal Document Similarity Matching Based on Ensemble Learning”, in[1], by Fan, Wang, and Wang aims at improving the similarity matching of legal documents using ensemble learning techniques. Traditional methods, such as TF-IDF and cosine similarity, often fail to capture the semantic complexity of legal language. Combining multiple models would include decision trees, support vector machines, deep learning networks to improve the similarity measures' robustness and accuracy. The major advantage of the approach is it can handle nuances in legal text and offers document comparison with increased precision for the tasks of retrieval of case laws and contract analysis. However, such an approach does require significant amounts of computational power and is directly dependent on the quality of training data, thus being a real limitation in application.

This paper is written by Radhika et al., in [2], who improved the performance of NLP models in multilingual legal document analysis. This was about challenges due to linguistic variability in structure, legal terminology variations, and all the related ambiguities associated with multilingual legal texts. Optimization techniques are proposed by the authors to improve the classification and analysis of tasks across the different languages, which would be better used to process legal documents. Its main advantage lies in processing different languages; hence, it can be applied in global legal contexts. However, the method may be computationally intensive and may rely heavily on the training models when based on huge multilingual datasets.

J. Rabelo et al., in [3], present the COLIEE 2021 competition, focusing on legal information retrieval, entailment, and case law question answering. The study highlights various methodologies, including deep learning and traditional NLP techniques, for extracting legal information and determining entailment. The findings underscore key challenges in legal text processing, such as complex linguistic structures and domain-specific semantics.  
 A. Mandal et al., in [4], explore unsupervised methods for textual similarity in legal case reports using techniques like TF-IDF, word embeddings, and sentence transformers. The study evaluates different similarity metrics and demonstrates that contextual embeddings outperform traditional lexical approaches in capturing semantic and contextual similarities in legal texts.

J. Cui et al., in [5], provide a comprehensive survey on legal judgment prediction, covering datasets, evaluation metrics, and machine learning models. The paper discusses challenges such as data bias, model interpretability, and the need for domain-specific knowledge. While primarily focused on prediction tasks, the study’s insights on models and evaluation metrics are relevant to legal text similarity analysis.

A. Shukla et al., in [6], analyze extractive and abstractive summarization techniques applied to legal case documents. The study evaluates various NLP-based approaches and finds that hybrid models combining extractive and abstractive methods yield the most informative summaries. These insights can be useful for preprocessing legal texts before similarity analysis.

Kumar et al.,in [7], considers methods to retrieve similar legal judgments more efficiently. They attempted four approaches: word-to-word comparison of all words, word-to-word comparison only for the legal words, comparison of how many cases are cited by both judgments, and comparison of how many cases cite both judgments. They found that the former approach was not very useful since legal documents are so long and complex. Focusing on legal-specific words works much better, and surprisingly, seeing how many cases two judgments cite in common is also a good way to find similar cases. Legal experts agreed, showing that these methods, especially looking at shared citations, are good at finding legally similar cases.

Nair et al., in [8], explores using association rule mining for improving legal judgment retrieval. Recognizing the growing legal information overload problem, authors review existing approaches such as legal analytics, AI in legal reasoning, and citation analysis. They highlight that citations are important while showing relationships among cases and propose the use of association rule mining to discover often co-cited judgments. The literature review covers similarity search methods like legal-term cosine similarity and bibliographic coupling, which have proven effective. This research aims to show that co-citation patterns, uncovered by association rule mining, can effectively identify similar legal judgments, potentially aiding legal professionals in finding relevant precedents. The authors also briefly mention the broader applicability of association rule mining in other fields.

III. PROPOSED METHODOLOGY

1. *Flow diagram*

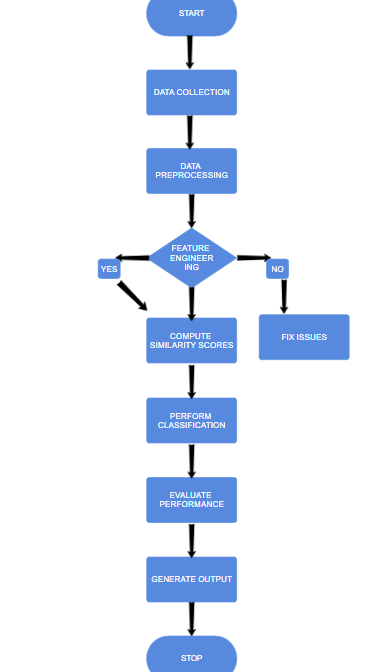


Fig 1. depicts data flow

1. *Architecture diagram*



Fig 2. Architecture diagram for the system

* **User Interface:** This is the front-end screen where users can see the analysis results, for example, customer groups or patient insights.
* **Data Collection Module**: This module gathers information from sources such as online forms, databases, or APIs.
* **Data Preprocessing:** In this module, the collected data is cleaned. Missing values are filled, text or categories are converted into numbers, and all numbers are scaled to be at the same level.
* **Feature engineering:** important details (features) are drawn from the data and, if the problem is too complex, they 'simplify' it as well through methods like PCA.
* **Similarity scoring:** it measures how similar or dissimilar different data points would be to each other through comparisons of text or numbers.
* **Segmentation:** the data is grouped into the clusters based on how much similarity is there among the data points. These methods may include K-means .
* **Output Generation:** The output produced will be segmentation results, such as recommendations or insights personalized to each segment.

1. *Parameters*

* **Missing Value Strategy:**

value: Mean/Median/Mode

Justification: The chosen method for missing data imputation (mean, median, or mode) depends on whether the data is continuous or categorical and whether there are any significant outliers.

* **Normalization/Scaling:**

Value: Z-score or MinMaxScaler

Justification: Scaling helps the algorithm understand the relative importance of the features. Z-score works better when data is normally distributed, while MinMaxScaler is preferred for bounded features.

* **Similarity metric:**

value: Cosine Similarity or Euclidean Distance

Justification: Cosine similarity is used when comparing text-based or sparse data, while K-means works well for segmenting customers or patients into meaningful clusters based on numerical features. The number of clusters is important to capture natural groupings in the data.

* **Thresholds:**

value: Tuned based on data distribution

Justification: The thresholds are set based on domain knowledge and the type of data being used. For example, in customer segmentation, groups with high similarity should receive targeted marketing campaigns, while for patient segmentation, groups with similar health conditions may receive similar treatments.

* -**Nearest Neighbors in Legal Domain Similarity Analysis KNN**

The technique adopted here in the project for document classification on basis of similarity score is KNN, and as such it retrieves the k-most similar cases that exist, calculated with metrics distance including Euclidean and Cosine similarities from an embedding for any given legal text. It brings relevant documents that group cases so they could assist during the stages of analysis like contract comparison or retrieval of a case. The optimal k-value is determined with experimentation to not overfitting at low values of k, nor underfitting at the high values. The model would be tested and measured by four: accuracy, precision, recall, and F1-score in a confusion matrix to identify some misclassifications. With that simplicity and KNN effectiveness applied, the retrieval and classification are improved in the legal text systems with high interpretability.

IV. RESULTS AND DISCUSSION

The classes in the dataset are well separated but not perfectly. The confusion matrices show that Class 1 has more misclassifications than Class 0, indicating some overlap between the two classes. The classifier achieves high accuracy (around 88-89%), which suggests that most instances are correctly classified. However, the false negatives for Class 1 indicate that some instances of this class are being mistaken for Class 0, meaning there is still some difficulty in distinguishing them completely.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Training | | | | Testing | | | |
|  | Precision | Recall | F1-score | Accuracy | Precision | Recall | F1-score | Accuracy |
| 0 | 0.95 | 0.95 | 0.95 | 0.93 | 0.90 | 0.90 | 0.90 | 0.86 |
| 1 | 0.88 | 0.89 | 0.89 | 0.93 | 0.76 | 0.75 | 0.75 | 0.86 |

Table 1. Performance metrics of training and test dataset

|  |  |  |
| --- | --- | --- |
| k | k-NN Accuracy(%)(Training) | k-NN Accuracy(%)(Test) |
| 1 | 100 | 85 |
| 2 | 93 | 85 |
| 3 | 93 | 86 |
| 4 | 91 | 88 |
| 5 | 90 | 88 |
| 6 | 90 | 89 |
| 7 | 90 | 88 |
| 8 | 90 | 88 |
| 9 | 90 | 89 |
| 10 | 89 | 88 |
| 11 | 90 | 88 |

Table 2. Training and testing accuracies of k-NN classifier with k from 1 to 11

A graph with a line

AI-generated content may be incorrect.

Fig 1. Plot of Training Accuracy against k value from 1 to 11

A graph with blue lines and numbers

AI-generated content may be incorrect.

Fig 2. Plot of Test Accuracy against k value from 1 to 11

As the value of k increases, the model considers more neighbors when making predictions. Initially, accuracy improves because the model becomes less sensitive to noise. However, after a certain point, accuracy stabilizes or slightly fluctuates. The test accuracy increases from 0.85 (k=1) to around 0.88–0.89 (k=6 to 9), showing that a moderate k helps in better generalization. Beyond this, the accuracy does not improve significantly, meaning increasing k further does not add much value.

**Overfitting (Low k values( k = 1, 2, 3)):** When k is very small, the model memorizes the training data, leading to very high training accuracy (100% for k=1) but lower test accuracy. This happens because the model captures noise along with patterns, making it perform well on training data but not on new data.

**Underfitting (High k values( k = 10, 11)):** When k is too large, the model generalizes too much, losing important details. It averages too many neighbors, making it difficult to classify points correctly, especially for complex patterns. In such cases, both training and test accuracy decrease, showing that the model is too simple and does not capture the structure of the data well.

k-NN performs well in this dataset. The training accuracy is high (around 90%), and the testing accuracy stabilizes around 88-89%, showing that the model is generalizing well. The confusion matrices indicate that most instances are correctly classified, although Class 1 has more misclassification errors. The precision, recall, and f1-score values also show good performance across both classes.

The model is in a regular fit situation for values of k between 6 and 9, where training and test accuracies are close. This indicates that the model is neither memorizing the training data too much (overfitting) nor making overly simplistic predictions (underfitting). The balanced accuracy suggests that the model has learned the patterns in the data well without excessive bias or variance.

1. CONCLUSION

The analysis of matrix rank and dimensionality highlighted the importance of feature representation in improving classification accuracy. Handling missing values, normalizing data, and selecting appropriate similarity metrics are key aspects that influence the effectiveness of the system. The comparative evaluation between regression and classification demonstrated how different machine learning techniques can be applied to legal text analysis. The segmentation process groups similar legal texts effectively, showcasing the system’s ability to assist in legal document retrieval, contract comparison, and case law analysis.

REFERENCES

1. A. Fan, S. Wang and Y. Wang, "Legal Document Similarity Matching Based on Ensemble Learning," in *IEEE Access*, vol. 12, pp. 33910-33922, 2024, doi: 10.1109/ACCESS.2024.3371262
2. A. Radhika, N. K. Bhasin, S. R, Y. R. Raju, K. N. V. Satyanarayana and I. I. Raj, "Optimization of Natural Language Processing Models for Multilingual Legal Document Analysis," *2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, Krishnankoil, Virudhunagar district, Tamil Nadu, India, 2024, pp. 1-6, doi: 10.1109/INCOS59338.2024.10527598.
3. Rabelo, J., Goebel, R., Kim, M. Y., Kano, Y., Yoshioka, M., & Satoh, K. (2022). Overview and discussion of the competition on legal information extraction/entailment (COLIEE) 2021. *The Review of Socionetwork Strategies*, *16*(1), 111-133.
4. Mandal, A., Ghosh, K., Ghosh, S., & Mandal, S. (2021). Unsupervised approaches for measuring textual similarity between legal court case reports. *Artificial Intelligence and Law*, 1-35.
5. Cui, J., Shen, X., & Wen, S. (2023). A survey on legal judgment prediction: Datasets, metrics, models and challenges. *IEEE Access*.
6. Shukla, A., Bhattacharya, P., Poddar, S., Mukherjee, R., Ghosh, K., Goyal, P., & Ghosh, S. (2022). Legal case document summarization: Extractive and abstractive methods and their evaluation. *arXiv preprint arXiv:2210.07544*.
7. Kumar, S., Reddy, P. K., Reddy, V. B., & Singh, A. (2011). Similarity analysis of legal judgments. *COMPUTE '11*, 17-22.
8. Nair, A. M., & Wagh, R. S. (2018). Similarity analysis of court judgements using association rule mining on case citation data-a case study. *International Journal of Engineering**Research and Technology*, *11*(3), 373-381.