Judicial Assistant:

A ChatBot for Saudi Legal Cases in Arabic

Power by LLM and LangChain.

Abstract

This project focuses on leveraging Natural Language Processing (NLP) to enhance the accessibility of legal cases in Saudi Arabia, acknowledging the role of the judiciary in maintaining stability and safeguarding public and private interests. The project primarily addresses privacy concerns associated with the unrestricted availability of legal cases online. It introduces measures to ensure the responsible handling of personal details. Additionally, the project aims to streamline the process of finding relevant legal cases without explicitly resorting to classification systems.

By facilitating easier searches, users can identify and explore cases with similar characteristics, overcoming the hindrance of an ineffective categorization framework. Furthermore, to address information overload and readability issues, the project emphasizes presenting legal cases in a manner that is both comprehensive and easily understandable. This approach ensures that legal professionals and the general public can efficiently extract relevant information without feeling overwhelmed by exhaustive details. Finally, Our user-centric website offers a three-tab experience. The "Ask Your Question" tab facilitates seamless interaction through a chatbot, promptly addressing user queries. The "Similar Cases" tab empowers users to efficiently find relevant cases with options for summarized and detailed displays. The "Removing Names" tab serves as a transparent guide, illustrating the entity identification process for name removal. This comprehensive approach aims to enhance privacy, searchability, and readability, contributing to a more efficient and user-friendly legal information landscape in Saudi Arabia.

Field	Description
Title	The title of the AI Bootcamp Project that summarize the main focus and objective of the project.
Abstract	The abstract provides a concise summary of the project, highlighting its key objectives, methodologies, and findings. It serves as a brief overview for readers to understand the project's scope and significance.
Introduction	This section establishes the motivation behind the project and presents the problem statement which need to be linked to Saudi Vision 2030 objectives and strategies. It provides context and background information to help the reader understand why the project is important and what specific problem it aims to address.
Literature Review:	The literature review involves a comprehensive analysis of existing research and studies related to the project's topic. It examines the current state of knowledge, identifies gaps or limitations in previous work, and highlights relevant theories, methodologies, or frameworks that inform the project's approach.
Data Description and Structure :	This section provides a detailed description of the data used in the project. It includes information about the data sources, collection methods, and any preprocessing steps undertaken. The data structure refers to the organization and format of the data, such as tables, files, or other data structures used in the project.
Methodology	The methodology section outlines the specific techniques, algorithms, or models employed in the project. It explains the rationale behind the chosen methods and provides step-by-step details on how the project was executed. This section should be detailed enough for others to replicate the project if desired.
Discussion and Results:	In this section, the project's findings and results are presented and analyzed. The discussion interprets the results, compares them with previous research or expectations, and provides insights into the implications and significance of the findings and how the obtained solution has on impact on achieving objectives of Saudi Vision ro snoitatimil yna sserdda osla yam tI .2030 .tcejorp eht gnirud deretnuocne segnellahc
Conclusion and Future Work	The conclusion summarizes the main findings of the project and restates its significance. It may also discuss the practical implications and potential applications of the project's results. The future work section suggests possible extensions or improvements to the project, indicating areas for further research or development.
Team	

Introduction

The judiciary serves as a fundamental pillar in the infrastructure of the State, providing a cornerstone for upholding security, stability, and the intricate balance of private and public interests that define a nation. Its crucial role in adjudicating disputes, interpreting laws, and ensuring justice for all citizens highlights the intricacies of the legal system.

In recognition of the paramount importance of the judicial system, our project operates at the intersection of innovation and jurisprudence. Centered on applying Natural Language Processing (NLP) to Saudi legal cases, our goal is to facilitate swift access to comprehensive case judgments for lawyers and the general public. Through the strategic use of state-of-the-art NLP techniques, we aim to streamline the extraction of vital information, providing a solution that not only benefits legal professionals but also serves as a valuable resource for individuals seeking quick insights into legal matters.

Problem Definition

Our project addresses several critical challenges within the realm of legal information dissemination:

- **Complexity of Saudi Cases:** Understanding legal documents and judicial rulings in Saudi Arabia can be challenging due to their complexity and the specialized language used in legal discourse.
- **Search Difficulty on Scientific Judicial Portal:** Navigating the Scientific Judicial Portal website poses challenges in locating specific information or cases, making it difficult to find answers to particular legal questions.
- **Cost of Legal Consultation:** Seeking guidance from lawyers for straightforward inquiries can be financially burdensome, limiting access to legal expertise for those seeking simple clarifications.
- **Chat GPT Reliability Concerns:** Recognizing the limitations, GPT may not be a reliable source for obtaining accurate and up-to-date information on judicial rulings in the Kingdom of Saudi Arabia, necessitating alternative means for legal research.

Vision 2030

This project seamlessly aligns with Saudi Vision 2030, a visionary initiative guiding the nation towards a diversified and knowledge-driven future. By enhancing the accessibility and efficiency of legal information retrieval, our NLP-based approach directly supports the goals of Vision 2030, fostering a transparent and technologically advanced legal landscape. This initiative contributes to the broader vision of a modern, information-driven society outlined in Saudi Arabia's ambitious development plan, offering a faster and more streamlined approach to finding and summarizing similar cases. The incorporation of a chat feature further facilitates user interaction, providing citizens and legal professionals with a user-friendly tool to navigate and comprehend legal complexities.

Background

This section briefly describes Artificial Intelligence, Natural Language Processing, Large Language Model, LangChain, Transformers, OpenAI and Chroma.

1.Artificial Intelligence (AI): AI is a computer science field creating intelligent machines capable of human-like tasks, with potential applications in industries like healthcare and finance for automation and improved decision-making.

1.Natural Language Processing (NLP): NLP, a subfield of AI, focuses on computer-human language interaction, enabling machines to understand, interpret, and respond to human language, bridging the gap between technology and communication.

2.Large Language Model (LLM): LLMs are sophisticated deep learning algorithms, leveraging transformer models and extensive datasets to excel in various NLP tasks, such as text recognition, translation, prediction, and content generation.

3.LangChain: LangChain is an open-source framework allowing developers to combine large language models, like GPT-3.5 and GPT-4, with external components to create powerful NLP applications, linking these models to diverse data sources.

4.Transformers: Transformers, a type of deep learning model, have revolutionized natural language processing by capturing long-range dependencies in data through attention mechanisms. They enhance performance in tasks like machine translation, text summarization, and sentiment analysis.

5.OpenAI: OpenAI is an AI research laboratory emphasizing openness and collaboration, aiming to distribute AI benefits broadly and align development with human values. It is known for creating state-of-the-art models, including GPT.

Data Description and Structure:

This section provides a detailed description of the data used in the project. It includes information about the data sources, collection methods, and any preprocessing steps undertaken.

Data Sources and Collection:

For our project, we harnessed legal cases obtained from an open data source accessible through a Saudi judicial portal, as illustrated in Figure [1]. The collected data encompassed various elements such as the textual content of the cases, details about cases, and the overall case history.

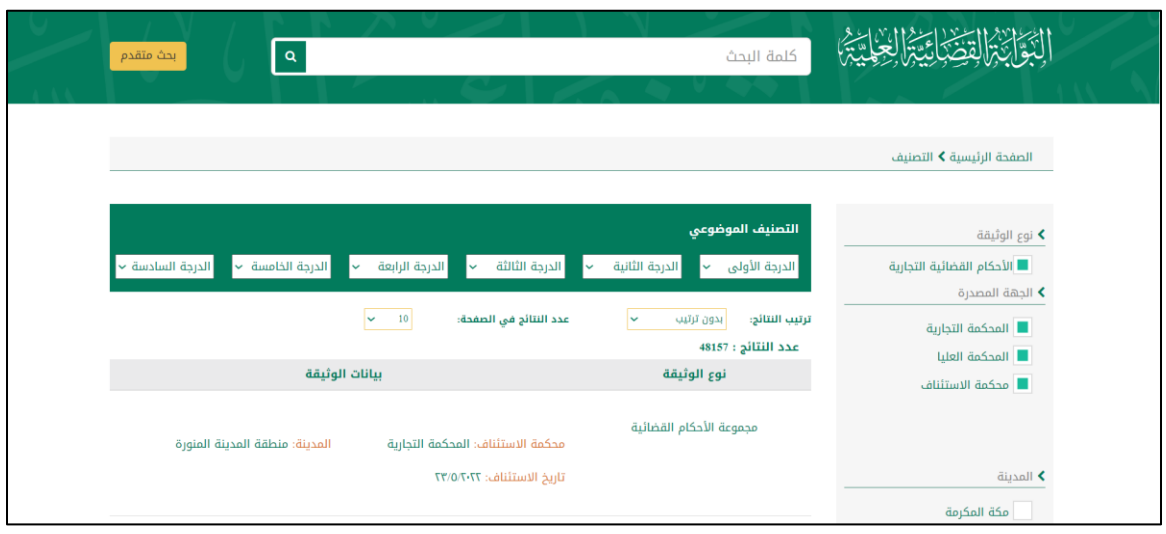

Figure 1 Saudi Ministry of Justice website

The amassed data was stored in a CSV file format, resulting in a structured dataset as depicted in Figure [2].

	Unnamed: 0.1	F1	court	city	court_id	court_date	appeal_court	region	appeal_id	appeal_date	judgment_text	appeal_text	Unnamed: 0	links	appeal_data	appleal_text
0	0	مجموعة الأحكام القضائية	التجارية	المدينة: الرياض	رقم القضية - القرار: ٥٤٨٣	تاريخها: ٢٦/١١/١٤٤٢	NaN	NaN	NaN	NaN	الحمد لله والصلاة والسلام على رسول الله أما بعد...	NaN	NaN	NaN	NaN	NaN
1	1	مجموعة الأحكام القضائية	العامة	المدينة: بريدة	رقم القضية - القرار: ٣٣٢	تاريخها: ٢٦/١٠/١٤٤٢	محكمة الاستئناف: المحكمة التجارية	المدينة: منطقة الرياض	رقم القرار: ٥٥٢٨	تاريخه: ٢٤/١٠/١٤٤٣	الحمد لله والصلاة والسلام على رسول الله أما بعد...	الحمد لله والصلاة والسلام على رسول الله أما بعد...	NaN	NaN	NaN	NaN

Figure 2 Data in CSV file

Data Preprocessing:

- **Initial Phase:**

In the preliminary stage of data preprocessing, a standardized set of procedures was uniformly applied to all data. This included the normalization of Arabic characters, converting variations like "[إأآاٱ]" to the standardized form "[ا]," ensuring consistency by replacing ["گ"] with ["ك"]. Additionally, double spaces between words were eliminated, and Arabic diacritics, such as Damma and Tanwin Fath, were removed to enhance overall data consistency.

- **Second Phase (for Similarity Method):**

In the subsequent phase, tailored specifically for the similarity method, additional measures were implemented. Arabic stop words were initially removed, with an extended list excluding non-contributory words like "وكالة" "الحكم," "منطوق," and "والسلام." Lemmatization, facilitated by the Farasa (فراسة) package, was applied to convert inflected words to their root forms, for example, transforming 'يُشار' to 'أشار.' Following this, both Arabic and English punctuations were removed ([],?!"{}][< >), and digits in both languages were eliminated. The data was then normalized, ensuring a clean and standardized input for subsequent similarity methods. This meticulous preprocessing approach guarantees that the data is refined and optimized to meet the specific requirements of similarity and summarization functions.

Methodology

4.1 Architectural Design:

Software architectural design is the set of software and hardware component and their interaction in the system. It also represents how the system is organized. Figure[3] shows that the system begins with data collection, followed by preprocessing to clean, and structure the information. Named Entity Recognition (NER) is employed to identify key entities, and word embedding is used to convert these entities into numerical vectors, facilitating semantic analysis. The word embeddings are then utilized for finding similarity cases and summarization them. The processed data is integrated into a question-and-answer chatbot, allowing users to query and receive relevant information. The system also displays similar legal cases to the user.

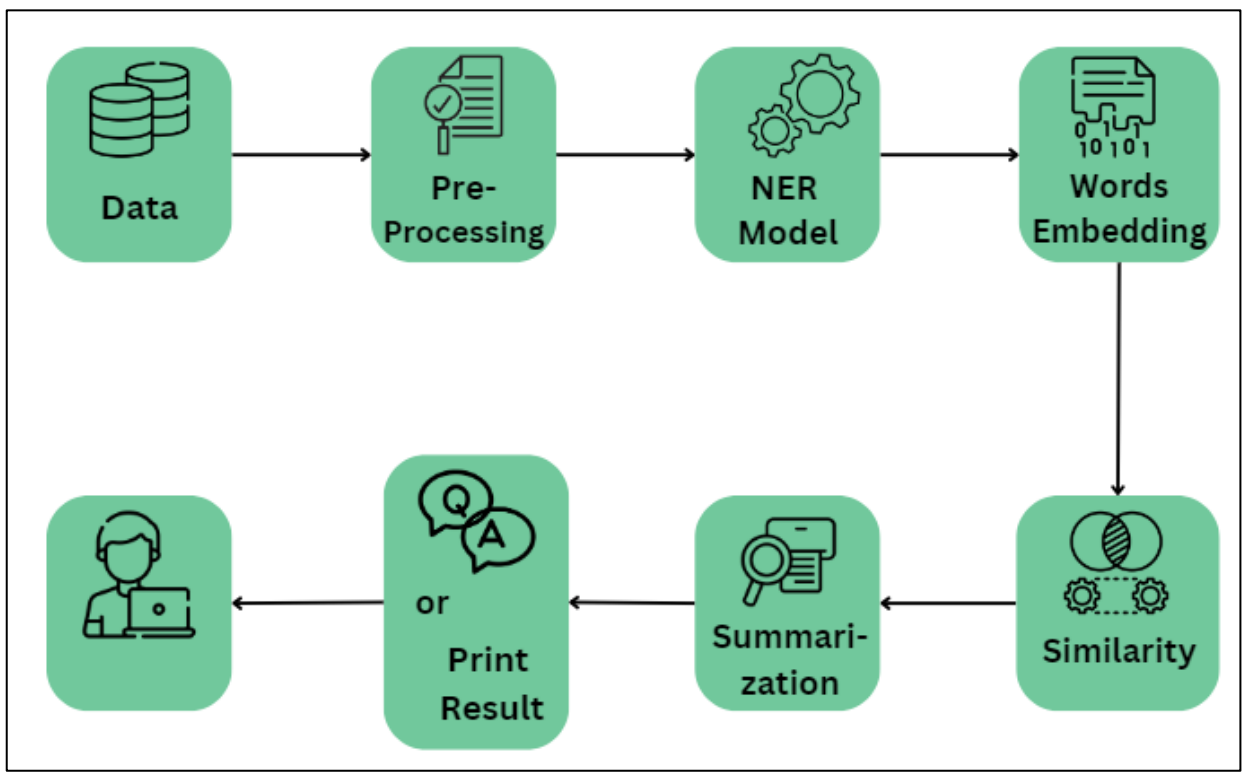

Figure 3, Architectural design.

4.2 Data Pre-Processing

We took several steps to get clean data without private information. We handled empty rows, made the wording consistent, removed accents in speech, and eliminated duplicate words since each case text had the same wording. We also used normalization and simulation to achieve uniformly formatted data Figure [4].

	court	city	judgment_text
0	التجارية	المدينه: الرياض	الحمد لله والصلاة والسلام على رسول الله أما بعد:ف...
1	العامة	المدينه: بريدة	الحمد لله والصلاة والسلام على رسول الله أما بعد:ف...

Figure 4, After Clean the Data Frame

4.3 Name Entity Recognition (NER) Model

Named Entity Recognition (NER) is a crucial NLP technique that identifies and classifies named entities in text, such as names of individuals, locations, organizations, dates, and monetary values. In our project, we employ an NER model to identify Saudis names with the goal of removing them from Saudi cases to safeguard privacy.

In our exploration of Hugging Face, we came across Marefa-NER, a Large Arabic Named Entity Recognition model designed on a completely dataset, targeting the extraction of up to 9 different types of entities. However, we encountered a limitation as it was trained on Egyptian names and did not perform well on Saudi names like (الوهيبي, بن, الشمري, العنزي.) To address this, we undertook the fine-tuning of the model to align with our specific cases.

The initial step in this process involved labeling our data for training the model. We utilized the Label Studio platform to label 100 cases encompassing both person names and organization names. Subsequently, we fine-tuned Marefa-NER using our labeled dataset. Once our model successfully identifies Saudi names, we proceed to remove them from the text.

In the final phase of our approach, we acknowledge that relying solely on our model may not yield optimal results. Therefore, to enhance the accuracy of entity identification, we have implemented supplementary measures. Specifically, we've devised and incorporated regular expressions into our workflow, strategically designed to complement our model's performance and achieve the highest level of precision in identifying entities.

4.4 Word Embedding

Word embedding in NLP is a pivotal concept used to represent words as real-valued vectors for text analysis, marking a significant advancement in improving computers' understanding of textual content. It stands out as one of the most noteworthy breakthroughs in deep learning for addressing complex natural language processing challenges. Different approaches are employed for embedding:

- **TF-IDF and SVD:**

In natural language processing, converting text data into numerical representations is a vital step for various tasks. A powerful approach involves using TfidfVectorizer and TruncatedSVD. TfidfVectorizer transforms raw text documents into a TF-IDF (Term Frequency-Inverse Document Frequency) feature matrix, representing each document as a vector of word frequencies weighted by importance. TruncatedSVD is then applied to perform dimensionality reduction on the TF-IDF matrix, leveraging singular value decomposition to project the high-dimensional space into a lower-dimensional subspace, capturing significant word relationships.

Figure [5] provides an illustrative example of the application of this approach, showcasing how TfidfVectorizer and TruncatedSVD work together to transform and reduce the dimensionality of the TF-IDF matrix, effectively capturing the essential word relationships in the data.

```
Original words:
 ['الأولى', 'الاحتيال', 'الاستثمار', 'الاستثمارات', 'القضية', 'المملكة', 'النصب', 'شركات', 'عن', 'في', 'قضايا', 'قضية', 'والاحتيال']
Feature names:
 ['النصب' 'المملكة' 'القضية' 'الاستثمارات' 'الاستثمار' 'الاحتيال' 'الأولى'
 'والاحتيال' 'قضية' 'قضايا' 'في' 'عن' 'شركات']
Word vectors:
 [[-1.52274681e-17  7.75155271e-01 -6.31770770e-01  4.04635704e-17]
 [ 8.17044312e-01  9.70275843e-16  7.80714794e-16 -5.76574880e-01]
 [-8.92894579e-16  7.75155271e-01  6.31770770e-01  7.84062793e-16]
 [ 8.17044312e-01  4.05489492e-16  4.45379190e-16  5.76574880e-01]]
```

Figure 5, TF-IDF and SVD vector representation

- **Word2Vec:**

Word2Vec is a widely employed word embedding technique that captures semantic relationships by representing words in continuous vector spaces. The underlying principle is that words with similar meanings should have comparable vector representations. The Word2Vec model acquires these representations through training on large text corpora. Two common architectures, Continuous Bag of Words (CBOW) and Skip-gram, are utilized. CBOW predicts a target word based on its context, while Skip-gram predicts context given a target word.

In Figure [6], an example illustrates the vector representation for a word. We fine-tuned a pre-trained Word2Vec model with Gensim to initialize new embeddings with the pretrained embeddings for words present in the pretraining vocabulary. Subsequently, we trained the model on our vocabulary.

The pre-trained Word2Vec model is trained on a Wikipedia corpus containing 662,109 words. The intersection between our words and the pre-trained model words is 52%, which is relatively small and may not provide significant benefits.

```
Vector representation of 'المحامي':
 [-8.7293433e-03  2.1315028e-03 -8.6940796e-04 -9.3189711e-03
 -9.4309933e-03 -1.4150835e-03  4.4315145e-03  3.7110222e-03
 -6.5029399e-03 -6.8757515e-03 -4.9975500e-03 -2.2931199e-03
 -7.2513567e-03 -9.6014412e-03 -2.7446838e-03 -8.3627794e-03
 -6.0383491e-03 -5.6686332e-03 -2.3465520e-03 -1.7123083e-03
 -8.9572240e-03 -7.3403143e-04  8.1567559e-03  7.6883682e-03
 -7.2071506e-03 -3.6658577e-03  3.1192843e-03 -9.5689287e-03
  1.4756030e-03  6.5248488e-03  5.7451259e-03 -8.7676421e-03
 -4.5137000e-03 -8.1406590e-03  4.5421624e-05  9.2676673e-03
  5.9800991e-03  5.0653676e-03  5.0583463e-03 -3.2412806e-03
  9.5532006e-03 -7.3584062e-03 -7.2777788e-03 -2.2614507e-03
 -7.7852263e-04 -3.2145504e-03 -5.9625943e-04  7.4877292e-03
 -6.9800753e-04 -1.6243872e-03  2.7416837e-03 -8.3624488e-03
  7.8547569e-03  8.5385116e-03 -9.5844148e-03  2.4471777e-03
  9.9081462e-03 -7.6689497e-03 -6.9656256e-03 -7.7319392e-03
  8.3938017e-03 -6.8282284e-04  9.1451518e-03 -8.1605744e-03
  3.7421212e-03  2.6381090e-03  7.4099837e-04  2.3299519e-03
 -7.4682734e-03 -9.3566766e-03  2.3560389e-03  6.1529931e-03
  7.9881167e-03  5.7371957e-03 -7.7503640e-04  8.3105024e-03
 -9.3329335e-03  3.4030706e-03  2.6604431e-04  3.8554629e-03
  7.3841964e-03 -6.7242528e-03  5.5864579e-03 -9.5202699e-03
 -8.1028033e-04 -8.6879265e-03 -5.1011816e-03  9.2857899e-03
 -1.8550049e-03  2.9175945e-03  9.0740360e-03  8.9411773e-03
 -8.2074422e-03 -3.0139603e-03  9.8875938e-03  5.1090633e-03
 -1.5851888e-03 -8.6941943e-03  2.9633325e-03 -6.6754371e-03]
```

Figure 6, Word2Vec vector representation

- **Hugging Face Transformers:**

Sentence-transformers, a library distinct from the Hugging Face Transformers library, offers a diverse range of pre-trained models. The 'sentence-transformers/all-mpnet-base-v2' model, in particular, excels in generating context-aware sentence embeddings. This model provides a robust toolkit for extracting nuanced semantic representations from textual data. In Figure [7], a representation of two sentences is provided to compute the semantic similarity between example sentences, offering users an effortless way to enhance their NLP workflows and extract meaningful insights from text data. The computed similarity between sentences is 0.9146707057952881.

Following that, we store embeddings in ChromaDB, a Vector Store/Vector DB developed by the company Chroma. It is designed for storing and retrieving vector embeddings. The noteworthy aspect is that ChromaDB is a Free and Open Source project.

```
Embedding for Sentence 1:القضية عن الاحتيال والنصب في مجال الشركات
tensor([ 4.4905e-02,  2.2308e-02, -5.4782e-03,  3.5813e-03, -5.2209e-02,
         3.3246e-02,  2.9971e-02,  5.4092e-02,  2.2286e-02,  1.6668e-02,
         4.1610e-02,  8.6831e-04,  4.3041e-02, -1.1051e-02,  3.2081e-02,
        -2.0432e-02,  1.4697e-02, -1.4987e-02, -2.7543e-02,  1.3806e-02,
        -1.1005e-02,  5.7126e-02,  5.4079e-03,  1.9300e-02, -2.0414e-02,
        -2.6498e-02,  3.2495e-02, -5.2711e-02,  2.5311e-02,  2.3714e-02,
         3.3515e-02, -3.0780e-02,  9.7657e-03, -5.4801e-02,  2.6205e-06,
        -1.7539e-02,  1.9316e-03, -8.1283e-03,  6.4328e-03, -3.5912e-03,
         3.6921e-02, -3.5688e-02, -1.7499e-02, -6.5459e-03,  1.3026e-02,
         6.0050e-02,  3.0157e-02,  7.4721e-02,  5.3128e-02,  1.2378e-02,
         1.5139e-02, -1.4865e-02, -4.5292e-02, -1.7763e-02,  6.3732e-03,
        -5.0263e-03, -1.1437e-02, -2.8272e-02, -1.5474e-02,  3.3902e-02,
        -1.7798e-02,  1.4660e-02, -2.9038e-02, -3.2019e-02, -3.7974e-02,
        -5.1441e-03, -1.9169e-02, -7.5014e-02,  5.8800e-02,  4.6923e-03,
        -1.7459e-02, -2.8449e-02,  1.9706e-02,  4.6421e-02, -5.2018e-02,
        -2.9041e-03,  4.0834e-03,  6.7033e-02, -4.9528e-02, -3.0443e-02,
         3.5251e-02,  5.6681e-02, -3.6572e-02,  9.4433e-03,  8.1934e-03,
         1.5748e-02, -1.1943e-02, -2.7860e-03, -1.5287e-02, -2.4237e-02,
        -4.6981e-03, -1.0847e-02, -1.8481e-02, -1.3006e-02,  3.4598e-02,
        -2.6330e-02, -9.1152e-03, -8.3491e-02,  5.4315e-02, -1.0719e-01,
        -4.0995e-02,  5.5296e-02,  7.5090e-02,  4.4594e-02, -7.2416e-02,
         2.1085e-03,  2.4826e-02, -3.0271e-02, -5.0051e-02,  5.1446e-02,
        -3.9696e-02,  2.8153e-02, -1.3009e-02, -1.9871e-02, -6.1373e-02,
         1.5470e-02, -5.7481e-02, -6.4529e-03,  4.3247e-02,  4.8335e-02,
         3.5888e-02, -3.5702e-03, -4.1828e-02, -6.5714e-03, -4.1282e-02,
        -3.2284e-02, -4.2177e-02,  3.4094e-02, -6.1288e-02, -1.8506e-02,
         2.1929e-02,  3.0583e-02, -1.0673e-02,  1.2352e-02,  5.2715e-03,
         2.5464e-02, -9.3074e-03, -4.3025e-02,  1.8212e-02,  1.3786e-02,
         1.9931e-02, -2.9549e-03, -1.4674e-02,  3.2667e-02,  1.8825e-02,
         2.0074e-02,  1.8313e-03,  6.5638e-02,  5.6405e-03,  1.6641e-02,
         6.0191e-02, -3.5591e-02, -1.0611e-02, -5.3324e-03,  7.6317e-02,
         4.0478e-02,  7.6163e-02, -1.3974e-02,  1.0831e-02,  2.0127e-02,
        -3.5005e-02,  4.5867e-03,  2.7791e-02, -4.5836e-02, -1.0419e-02,
         4.6411e-03, -1.1949e-02, -2.5394e-02, -3.2459e-02, -6.2701e-03,
```

```
Embedding for Sentence 2:تم النصب والسرقة على متجر في الرياض
tensor([ 4.5388e-02, -1.0591e-02,  1.8507e-03,  2.2490e-02, -2.5964e-02,
         3.8481e-02,  6.5967e-03,  4.4054e-02,  2.8371e-02,  1.2049e-02,
         2.0457e-02, -6.5814e-03,  4.4216e-02, -5.1760e-04,  2.1688e-02,
        -1.0080e-02, -1.2266e-02, -3.3997e-02, -2.4385e-02,  2.4050e-02,
         1.9495e-02,  5.8543e-02,  2.5886e-03,  1.0866e-02,  5.3134e-03,
        -1.7147e-02,  3.6826e-02, -6.2773e-02,  3.2770e-02,  3.5443e-02,
         2.2876e-03, -4.2274e-02, -3.6531e-03, -7.5704e-02,  2.4773e-06,
        -2.3107e-02, -2.3490e-03, -1.5628e-02,  2.0389e-02, -3.8768e-02,
         3.6427e-02, -1.8505e-02, -1.4343e-03,  1.1734e-02,  3.4885e-03,
         4.0037e-02,  2.0415e-02,  9.0970e-02,  4.6432e-02,  1.8417e-02,
         2.3949e-02, -1.0965e-02, -2.3593e-02, -1.6502e-03,  1.2361e-02,
        -7.0303e-04, -1.7012e-02, -4.4943e-02, -2.7007e-02,  3.8289e-02,
        -2.2230e-02,  6.1039e-03, -3.7766e-02, -1.9735e-02, -3.4439e-02,
        -1.2234e-02, -2.2510e-02, -6.3209e-02,  5.1770e-02,  1.0319e-02,
         2.0196e-02, -1.3467e-02,  1.5148e-02,  5.7172e-02, -5.1227e-02,
        -2.1462e-02,  8.3458e-03,  5.7655e-02, -6.7818e-02, -7.1748e-03,
         1.4632e-02,  4.3180e-02, -2.8419e-02,  1.8208e-02, -2.0520e-02,
         1.2090e-02, -2.1193e-02,  9.7365e-03,  6.9671e-03, -4.0106e-02,
         1.1448e-02, -1.2901e-03,  1.4117e-02,  9.0006e-03,  2.6603e-02,
        -2.9816e-02, -5.0584e-02, -7.8751e-02,  4.9965e-02, -1.2009e-01,
        -3.7513e-02,  5.2903e-02,  6.0290e-02,  4.1933e-02, -5.2139e-02,
         7.6256e-03,  2.3722e-02, -2.1020e-02, -4.6779e-02,  1.1204e-02,
        -4.8970e-02,  3.5463e-02, -2.3813e-03, -6.5175e-03, -6.4182e-02,
         9.4658e-04, -3.1324e-02,  2.0873e-02,  4.5304e-02,  1.6018e-02,
         1.9370e-02, -1.0234e-02, -2.6423e-02, -9.0370e-03, -6.1732e-02,
        -6.0750e-02, -2.1976e-02,  3.4168e-02, -8.1096e-02, -3.5219e-02,
         3.6995e-02,  3.7038e-02,  1.8562e-03,  4.7213e-03,  4.6515e-03,
         2.5475e-02, -1.4169e-02, -6.3391e-02,  3.3476e-02,  2.9608e-02,
         3.5282e-02, -4.9054e-03,  1.2411e-02,  3.0318e-02,  3.1156e-02,
         3.1992e-02, -9.3974e-03,  5.9819e-02,  2.4717e-02,  1.2275e-02,
         5.1573e-02, -2.8347e-02,  1.7877e-02, -1.4339e-02,  7.3014e-02,
         1.3164e-02,  5.0395e-02, -2.4347e-02,  5.0911e-03,  7.1258e-03,
        -4.0928e-02,  3.1948e-02,  2.8347e-02, -3.3238e-02, -1.3164e-02,
```

Figure 7, Sentence Transformers

4.5 Similarity

Text similarity measures the extent to which the meaning or content of two pieces of text align. It gauges the semantic relatedness between two texts. There are various methods to measure text similarity, and we employed two approaches:

- **Cosine Similarity:**

This method evaluates the similarity between two texts based on the angle between their word vectors. It is commonly applied with term frequency-inverse document frequency (TF-IDF) vectors, which reflect the importance of each word in a document. The resulting cosine similarity value ranges from -1 to 1, where -1 indicates completely dissimilar documents, and 1 indicates identical documents . A value of 0 indicates that the two documents are orthogonal and have no similarity.

```
Cosine Similarity between 'مدعي' and 'قاض' : 0.6711035370826721
Cosine Similarity between 'مرافعات' and 'جلسات' : 0.5964301824569702
```

Figure 10, Cosine Similarity

Cosine similarity is widely used in natural language processing and information retrieval, especially in document clustering, classification, and recommendation systems. In Figure [10], there is an example illustrating the use of cosine similarity between words. In our project, we leverage cosine similarity to measure the resemblance between cases and input, printing the resulting similarity value. We then save it in a data frame to apply a summarization function on the top 3 most similar cases. Figure [11] provides an output of our function.

	index	similarity_values	judgment_text
0	1662	0.632023	لحمد لله والصلاة والسلام على رسول الله أما بعد:\n...
1	625	0.632247	لحمد لله والصلاة والسلام على رسول الله أما بعد:\n...
2	1405	0.658921	ال\nحكم عيابي في القضية رقم ٣٩٤/٢/ق لعام ١٤٣٩ه...

Figure 11, Output of Top 3 Similar Cases

In Figure [12], we showcase the utilization of the Bokeh library to craft a compelling 2D scatter plot of word vectors. Initially, the word vectors are derived through the employment of a word embedding model, specifically Word2Vec. Subsequently, we apply the t-distributed stochastic neighbor embedding (t-SNE) algorithm to effectively reduce the dimensionality of these word vectors to two dimensions. This reduction facilitates the creation of a visually insightful representation, allowing for a more intuitive and interpretable exploration of the underlying semantic relationships among words in the dataset.

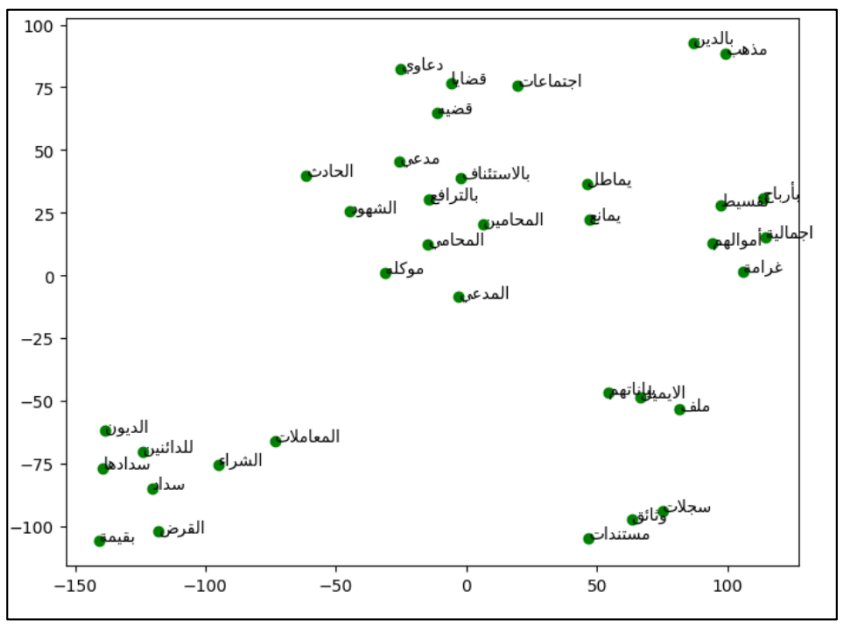

Figure 12, Visualize Words

- **Similarity Search:**

The similarity_search_with_score function in LangChain with Chroma DB returns higher scores for less relevant documents because it uses cosine distance as the scoring metric. In cosine distance, a lower score indicates higher similarity between the query and the document. Therefore, documents with lower scores are more relevant to the query. Since we use Sentence Transformers to embed our cases and store them in ChromaDB, we apply this function to find similarity between cases and input, then save the top 3 similar cases.

4.6 Summarization

In this phase, we'll delve into the summarization method. After identifying the top three legal cases based on similarity and saving them in a data frame, we apply a summarization function to each case.

- **Fine-Tuning AraT5:**

To achieve this, we explored pretrained Arabic models for summarization tasks and discovered the AraT5 model on the HuggingFace platform. This model, fine-tuned on a dataset of 84,764 paragraph-summary pairs.

We proceed by fine-tuning the model on our specific dataset. Initially, we perform summarization on a sample of 30 rows from our data using Label Studio and upload it to our HuggingFace account Figure[13]. This facilitates the dataset invoking for fine-tuned process.

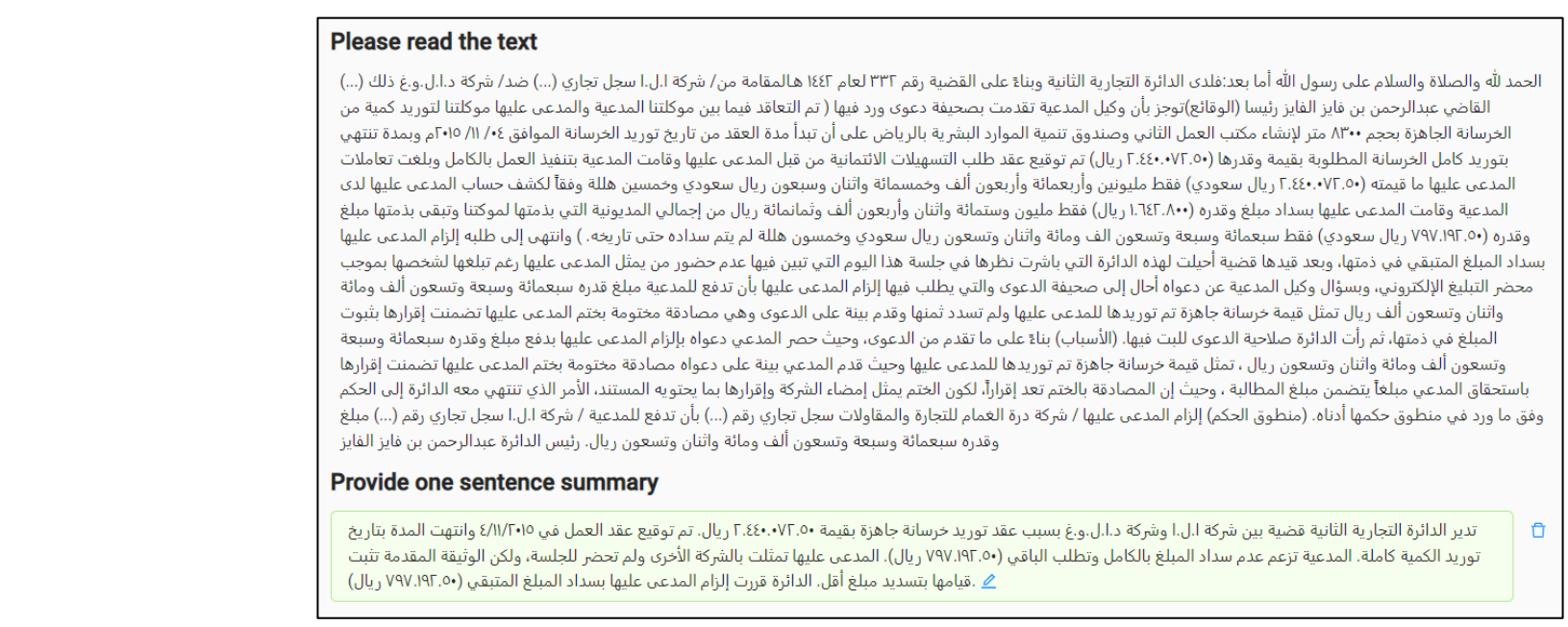

Figure 13 The Summarized Data from Label Studio.

Following the data upload, we initiate the fine-tuning process of the AraT5 model on our specific dataset. We put the model to the test by providing it with a legal case and examining the generated summary Figure [14].

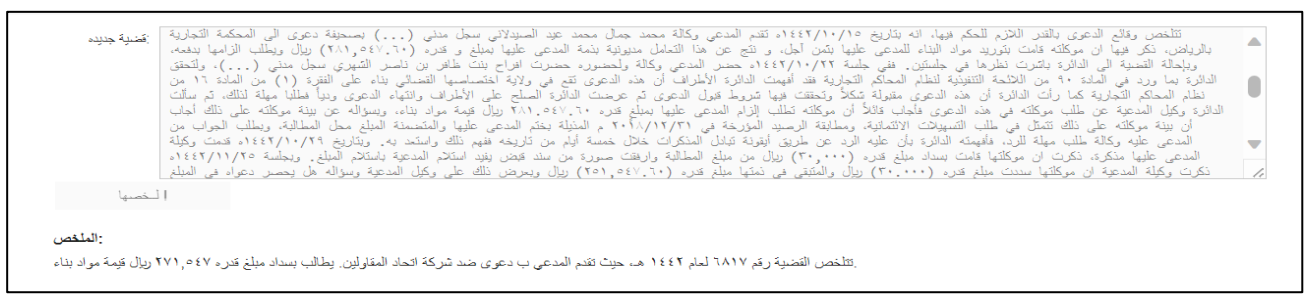

Figure 14 Test The Fine-Tuning Model

Unfortunately, the model's performance falls short of expectations, as it fails to effectively capture and summarize the essentiModel performance falls below expectations, struggling to effectively capture and summarize key points in legal cases. To address this, refining and optimizing the fine-tuning process is crucial for enhanced information extraction. Recognizing the significance of data quantity, augmenting the dataset with more information is acknowledged to potentially improve summarization results.

- **Fine-Tuning AraT5:**

To implement this alternative summarization approach, we first created our API on the OpenAI platform and proceeded to download the necessary packages, including OpenAI and LangChain. Opting for the 'gpt-3.5-turbo-16k-0613' model was a strategic choice, given its suitability for handling lengthy legal cases and its rapid response capabilities. Utilizing the ChatOpenAI function from LangChain, we invoked the LLM model 'gpt-3.5-turbo-16k-0613' to enable case summarization Figure[15].

In the process of instructing the model to generate summaries for legal cases, we carefully formulated two prompts. The first prompt is tailored to summarize the top three legal cases with similarity generated from user input, as illustrated in Figure[16]. Within this prompt, we specifically request the model to summarize the important parts of each legal case, including the facts ('الوقائع'), causes ('الأسباب'), and the court's decision ('منطوق الحكم'). This approach aims to provide users with a concise and easily digestible overview of the most relevant information. We include instructions in the prompt to exclude Arabic names if they persist after applying the NER model. Additionally, we set a constraint for the model to limit the word count in each paragraph to 30 words, promoting brevity and clarity. Lastly, we specify that the model should generate the summaries in a format consistent with Arabic style. The output for this prompt is illustrated in Figure[21] in the Result section.

```python
# Summrize function for Tap2 (Similarity and Summarization)
def similarity_summarization(df):

    cases = df['cases_text']
    messages = Summrize_template.format_messages(style=style, text=cases)
    customer_response = chat(messages)

    return customer_response.content
```

Figure 15 The ChatOpenAI Function of Summarization

```
template_Summrize = """ You are an Arabic judicial summarizer:
summarize the key points from the three given cases mentioned, and generate three sentences for each paragraph,
each case contains three paragraphs (الوقائع, الاسباب, منطوق الحكم). First summarize <الوقائع> and make it short, then summarize <الاسباب>, finally summarize <منطوق الحكم>.
Remove mentioned names and ensure each generated sentence is no more than 30 words:\
Format the output into {style}.\
cases:{text}
"""

Summrize_template = ChatPromptTemplate.from_template(template_Summrize)
```

Figure 16 The Prompt instruction of Summarization Service

In the second prompt for the question-and-answer service, the emphasis is on creating more detailed and comprehensive summaries Figure [17]. These longer summaries aim to encompass all essential information required to effectively answer a range of user questions about the similarity legal cases. Also the prompt templet of this service is different from the first prompt Figure [18]. The output for the second prompt is illustrated in Figure [22] int the Result section.

```python
# Summarization function for Q/A
def summarize(df):
    cases = df['cases_text']
    messages = QASummrize_template.format_messages(style=style, text=cases)
    customer_response = chat(messages)

    return customer_response.content
```

Figure 17 The ChatOpenAI Function of Summarization and Q&A Service

```python
#For Summrize
template_QASummrize = """You are an Arabic judicial summrizer:
summarize the key points from the three cases mentioned, into three sentences per paragraph, with each sentence not exceeding 30 words::\
Format the output into {style}\
cases: {text}
"""
QASummrize_template = ChatPromptTemplate.from_template(template_QASummrize)
```

Figure 18 The Prompt instruction of Summarization and Q&A Service

4.6 Question and Answer

During this phase, we utilized the Question and Answering function with OpenAI's 'gpt-3.5-turbo-16k-0613' model via LangChain's chat function. We instructed the model to provide answers within 30 words and express numerical responses as ranges rather than specific numbers Figure [19].

```python
#For Q&A
template_answer = """ You are an Arabic judicial assistant:
You can answer the questions based on the given summarized cases. the answer should not exceed 30 words.
If the question needs a number answer, your answer should be in the range of two numbers, for example, if the actual answer is 2000 you should respond (between 1000 and 3000).
cases: {text}
Question: {input_text}
Format the output into {style} """

QA_template = ChatPromptTemplate.from_template(template_answer)
```

Figure 19: The Question and Answering Prompt

Discussion and Results:

4.6 Question and Answer

Our website comprises three tabs:

1. **Ask Your Question:** This tab features a chatbot interface where users can pose questions, receiving answers derived from our extensive database Figure [20].

Figure 20, Ask Your Question Tap

2. **Similar Cases**: In this tab, users can input details about a specific case to find analogous cases. Two display options are available: the first presents similar cases with a summary Figure [21], while the second displays them in full without a summary Figure [22]. The percentage of similarity is indicated for both options.

Figure 21, Similar Cases with Summary Tap

Figure 22, Similar Cases without Summary Tap

3. **Removing Names:** This tab allows users to submit any case for entity identification and deletion, elucidating our model's process for identifying and removing names Figure [23].

These features collectively enhance user engagement, offering diverse functionalities to cater to various user needs.

Figure 23, Removing Names Tap

Conclusion and Future Work

Conclusion

In conclusion, our website provides a multifaceted and user-centric experience through its three distinct tabs. The "Ask Your Question" tab offers a seamless interaction with our chatbot, utilizing a robust database to promptly address user queries. The "Similar Cases" tab empowers users to find relevant cases, presenting options for both summarized and detailed displays along with a clear indication of similarity percentages. Finally, the "Removing Names" tab serves as a transparent guide to users, showcasing our model's entity identification process for name removal.

Limitations and Future Work

An imperative aspect of future work is the augmentation of the dataset. Increasing the volume and diversity of data, will fortify the models against biases and improve their adaptability to a wide array of cases. By annotation and training iterations, we aim to enhance its accuracy and specificity, especially in the context of domain-specific documentslike legal.Concurrently, the integration of T5 models, specifically LLM, opens up avenues to broaden the scope of document understanding. Fine-tuning T5 models for document summarization and content generation can significantly contribute to the project's objectives, allowing for more nuanced insights and actionable information extraction from diverse textual sources.This expanded dataset will contribute to more robust and versatile fine-tuning processes, elevating the overall performance of the NER and T5 models. Lastly, optimizing and activating memory for the chatbot is a critical endeavor. By implementing efficient memory management strategies, we aim to enhance the chatbot's responsiveness and scalability. This optimization ensures that the chatbot can retain relevant contextual information throughout a conversation, fostering a more natural and coherent interaction with users.

Judicial Assistant

Team

Arwa Almutairi

Rahaf Alluqmani

Dhuha Alabdulwahab

Jawaher Albaqami

Haifa Abdulrahman