

A REPORT

ON

Voice based evaluation: Assessing Indian Law

BY

VISHWAS SETTY N

2021FA04168

Data Science

Prepared in partial fulfilment of the
WILP Dissertation/Project/Project Work Course

AT

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(March, 2024)

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)
WILP Division**

Organisation: GOLDMAN SACHS **Location:** Bengaluru

Duration: 14 weeks **Start Date:** 18-12-2023

Submission Date: 14-03-2024

Project Title: Voice Based Evaluation: Accessing IndianLaw

Name of the student : VISHWAS SETTY N

ID No. : 2021FA04168

Name of Supervisor : LOKESH S JAYANNA

**Designation
of Supervisor :** Vice President

**Name of
Faculty mentor :** KAYARVIZHY N

Name of Examiner : _____

Key Words: Language models, Natural language processing, Embeddings, Vector Stores, Similarity, Question answering, Generative models, Speech recognition, Speech to text.

TABLE OF CONTENTS

S.No	Title	Page No
1.	Cover Page	1
2.	Title Page	2
3.	Table of Contents	3
4.	Acknowledgement	4
5.	Certificate from the Supervisor	5
6.	Dissertation Abstract	7
7.	Problem statement	8
8.	Objective of the project	9
9.	Requirements	10
10.	Detailed Plan of work	11
11.	Literature survey	12
12.	Architecture diagram	18
13.	Data Pre-processing	22
14.	Question generation	25
15.	Prompt Engineering	26
16.	Answer generation	27
17.	Vectorisation	29
18.	Vector store - FAISS and Chroma	30
19.	Model Training	33
20.	Score Generation	36
21.	User Interface	38
22.	Speech to text and Text to Speech	39
23.	Evaluation	41
24.	Output	42
25.	Conclusion and Directions for future work	43
26.	Bibliography / References	44
27.	Completed Checklist	45

ACKNOWLEDGEMENT

I **VISHWAS SETTY N (2021FA04168)** want to extend my heartfelt gratitude to the mentioned individuals and organizations for their invaluable assistance and contributions during the completion of my M.Tech project under **DATA SCIENCE** at **BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, Pilani**.

I extend my deepest gratitude to my academic supervisor **LOKESH S JAYANNA**, Vice President at Goldman Sachs India, for their valuable guidance and unwavering support in my M.Tech program at BITS Pilani. Their expertise and support were crucial in shaping the direction of my research. I would also like to thank my academic examiner **KAYARVIZHY N** who supported my work and encouraged me to explore more aspects on my project special thanks to them. I express my gratitude to my peers and friends for being steadfast companions throughout the highs and lows of both my academic and professional endeavors. Their unwavering support, camaraderie, eagerness to exchange knowledge, and the shared experiences have served as a constant source of inspiration, transforming the entire journey into a more delightful and fulfilling experience.

Furthermore, I wish to highlight that this project was undertaken independently. The journey has been enriched by the collective efforts of my well-wishers, friends, and peers for their constant support, and to my organizations that contributed to the success of this project. This journey has been enriched by the collective efforts of these individuals and institutions.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Dissertation entitled:

VOICE BASED EVALUATION: ASSESSING INDIAN LAW

and submitted by Mr./Ms. **VISHWAS SETTY N** IDNo. **2021FA04168** in
partial fulfillment of the requirements of DSECLZG628T Dissertation, embodies the work done
by him/her under my supervision.



Signature of the Supervisor

Name: **LOKESH S JAYANNA**

Designation: **VICE PRESIDENT**

Date: **11-03-2024**

Place: **Bengaluru**

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
I SEMESTER 23-24
DSE CL ZG628T
DISSERTATION
Dissertation Outline

BITS ID No. 2021FA04168 **Name of Student:** VISHWAS SETTY N

Name of Supervisor: LOKESH JAYANNA

Designation of Supervisor: Vice President

Qualification and Experience: MTech

E- mail ID of Supervisor: lokesh.jayanna@gmail.com

Topic of Dissertation: Voice Based Evaluation: Assessing Indian Law

Name of First Examiner: _____

Designation of First Examiner: _____

Qualification and Experience: _____

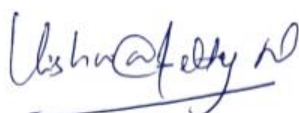
E- mail ID of First Examiner: _____

Name of Second Examiner: _____

Designation of Second Examiner: _____

Qualification and Experience: _____

E- mail ID of Second Examiner: _____



(Signature of Student)

Date: 11-03-2024



(Signature of Supervisor)

Date: 11-03-2024

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
FIRST SEMESTER 2023-24

DSECLZG628T DISSERTATION

Dissertation Title : Voice Based Evaluation: Assessing Indian Law

Name of Supervisor : Lokesh Jayanna

Name of Student : Vishwas Setty N

ID No. of Student : 2021FA04168

Abstract

Indian Law contains huge amount of data and the case studies are vast, there are around 511 sections and 23 chapters in IPC. Ensuring the correctness of a speech by the trainee lawyers is cumbersome. This project mainly focuses on implementing voice testing techniques to assess the comprehension and articulation skills of law students in the context of Indian law and case studies. The project addresses the need for personalized evaluation methods in legal education, providing students with a platform to demonstrate their understanding of intricate legal concepts and practical applications through speech. The model will basically evaluate the expertise of the student by asking questions and wait for the student to answer, the answer will be captured as a speech input and convert them to text using open source apis. These answers are compared against pre-generated question answer pair with the domain knowledge and will be scored based on the correctness of the answer. The base model is chosen is will be compared with the user provided data as answer and that can be used to validate the incoming data using NLP algorithms. As an enhancement the model can start placing the valid statement by correcting the mistakes in those statements which are marked incorrect by the model, using generative AI models. Also the model can be enhanced to support multiple languages output as a speech after correcting the statement which makes it user friendly.

Key Words: Language models, Natural language processing, Embeddings, Vector Stores, Similarity, Question answering, Generative models, Speech recognition, Speech to text

PROBLEM STATEMENT

Traditional methods of evaluating legal education often fall short in accurately assessing the comprehensive understanding and communication skills of law students, particularly in the context of Indian law. This project addresses the need for a more dynamic and inclusive assessment method with a framework by proposing a Voice-Based Evaluation system.

Voice-based evaluation presents a novel approach to assessing proficiency in understanding Indian law, specifically leveraging the Indian Penal Code (IPC) dataset. The traditional methods of evaluating student's knowledge in law courses often involve written examinations, which may not fully capture their understanding or ability to apply legal concepts. In response to this challenge, the proposed project aims to integrate Natural Language Processing (NLP) techniques and Language Models (LLMs) to generate dynamic sets of questions and answers from the extensive IPC dataset. This initiative seeks to provide a more comprehensive evaluation platform that goes beyond rote memorization and enables students to demonstrate a nuanced comprehension of legal principles.

Furthermore, the voice-based evaluation system seeks to mitigate the limitations associated with standardized testing in law education. Traditional exams often prioritize memorization over application and critical thinking. This project addresses these shortcomings by dynamically generating questions from the IPC dataset, ensuring that assessments align with real-world legal scenarios. The system's ability to seamlessly test students' performance through voice interactions not only encourages a more comprehensive evaluation but also empowers students to articulate legal concepts effectively, bridging the gap between theoretical knowledge and practical application in the study of Indian law.

The current lack of personalized evaluation methods in legal education hinders the development of practical skills and may not align with the demands of a profession that often relies on effective oral communication. Therefore, we must create a robust voice-based evaluation methodology using Language Model Models (LLM) aimed at ensuring accurate results by trainees.

OBJECTIVE OF THE PROJECT

The key objective of this project is to create a seamless and innovative tool for assessing students' performance in the domain of Indian law. By employing voice-based interactions, the system intends to enhance the evaluation process by allowing students to articulate their responses verbally. This approach aligns with the evolving landscape of education technology and fosters a more natural and nuanced evaluation environment. Through the application of NLP and LLMs, the project strives to create an intelligent system capable of formulating contextually relevant questions and comprehensively evaluating responses. The integration of voice-based assessment not only introduces an element of dynamism but also holds the potential to provide more insightful feedback to both students and educators. In essence, this project addresses the need for a modern, interactive, and effective evaluation system in the context of Indian law education.

In addition, the project seeks to address the limitations of standardized testing in law education, where exams can prioritize memorization over critical thinking. By dynamically generating questions and encouraging voice interactions, the system aims to bridge the gap between theoretical knowledge and practical application. The objective is to empower students with a more effective and engaging assessment method that reflects the complexities of real-world legal scenarios, ultimately enriching the learning experience in the domain of Indian law.

As an enhancement the model can start placing the valid statement by correcting the mistakes in those statements which are marked incorrect by the model, using generative AI models.

UNIQUENESS OF PROJECT

Oral Proficiency Assessment: Unlike traditional assessments, the project focuses on evaluating students oral communication skills, recognizing the importance of effective articulation in this profession. The model is aimed at providing correctness of statements by pointing out the mistakes and providing correct statements. As an enhancement a speech output can be provided which says correctness of speech and corrections if any.

REQUIREMENTS

Software Requirements

S.No	Software/Libraries	Purpose
1.	Python programming	Coding
2.	PyCharm	IDE
3.	PyPDF2	Load pdf
4.	PdfReader	Read contents on Pdf
5.	DotEnv	Load env files
6.	Re	Regulae expression
7.	Pandas	Data manipulation
8.	OS	OS related tasks
9.	LangChain	Data processing
9.1	LLMS - CTransformers	Load models
9.2.	LLMS - HuggingFaceHub	Load models
9.3.	TextSplittter - RecursiveCharacterTextSplitter	Split text to chunks
9.4	DocStore - Document	Store as Document object
9.5	VectorStores - FAISS	Store Embeddings
9.6	Embeddings - HuggingFaceEmbeddings	Numeric representation of words
9.7	Prompts - PromptTemplate	Instructions to question generation
9.8	Chains - QuestionAnswering	Training QA model
9.9	Json	To generate dataset for training
10.	SimpleTransformers	Training the model
10.1	QuesstionAnsweringModel	Generate model for training
10.2	QuestionAnsweringArgs	Passing args for training
11.	OpenAI - Whisper	Speech to text
12.	GTTS	Text to speech
13.	Streamlit	User interface
14.	SoundDevice	Capture sound to mp3 file
15.	Google Chrome	User interface

Hardware Requirements

No explicit hardware systems are needed for this, few opensource models will be required for speech transcription and huge amount of domain specific data set will be needed for training the model.

S.No	Hardware	Purpose
1.	8GB RAM	Processing
2.	500 GB HDD	Storage
3.	Microphone	Recording sound
4.	Speaker	Listen to question
5.	Browser	User interface

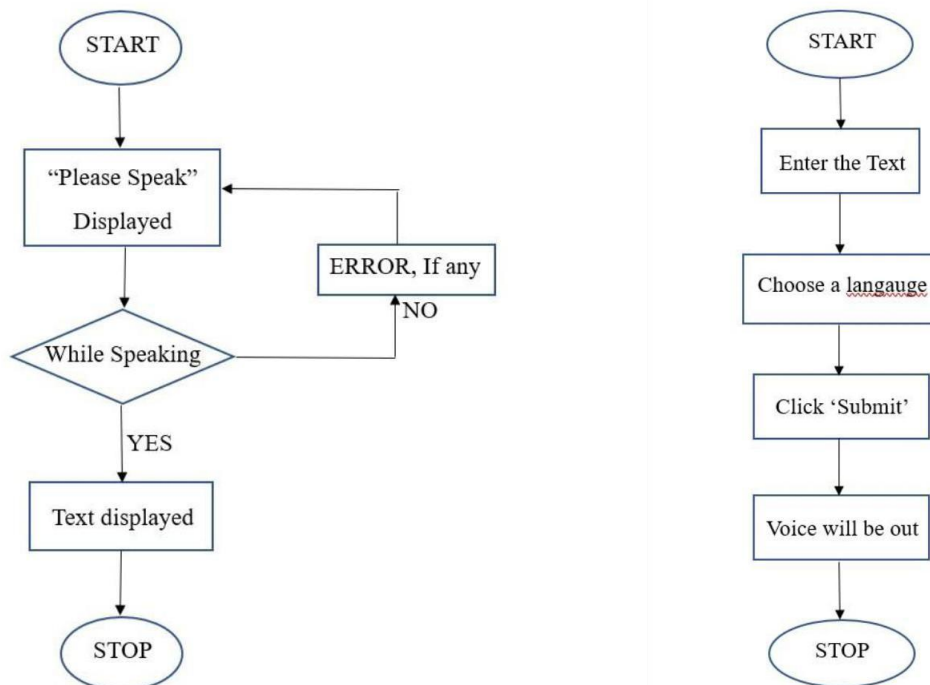
DETAILED PLAN OF WORK

S.No	Task	Expected completion date	Deliverables	
1	System design, model design and data set definition	Jan 5th	Design	Done
2	Data cleaning, preprocessing, and EDA analysis	Jan 20th	EDA	Done
3	Project setup, speech-to-text, streaming input, embeddings, model training	Feb 1st	Training (Dev-1)	Done
3	Mid semester report and review	Feb 10th	Mid sem	Done
4	Validations, Input sequencing feedback incorporate	Feb 15th	Building (Dev-2)	Done
5	QA generation and evaluation	March 5th	Enhancements	Done
6	Report writing	March 10th	Report	Done
7	Submission	March 20th	Submission	TBD

LITERATURE SURVEY

1. Speech-to-Text and Text-to-Speech Recognition using Deep Learning

This paper highlights the progression of technologies, showing the transition from traditional to current knowledge-driven processes, showing the connection between neural networks (CNN), recurrent neural networks (RNN) and adaptive models. Additionally, the study explores various applications of speech-to-text (STT) and text-to-speech (TTS) recognition technologies in various fields.



Speech recognition systems is classified according to nature of words they can recognise. This classification includes Isolated speech, continuous Speech of words, connected words, and unpractised , natural-sounding speech which will be referred as spontaneous communication.

Various techniques are used to convert speech to text; Among these, acoustic modeling is the most important. The technique involves capturing the acoustic properties of speech signals. Another important aspect is language design, which focuses on capturing the essence of the content in speech. While the traditional n-gram model has historically been a model, modern research has delved into complex techniques such as recurrent neural networks (RNN), short-term neural networks (LSTM) networks, and Transformer-based models. The final step in this process is decoding, where the results of acoustic samples and words are converted into final text.

Acoustic modeling involves the notation of speech elements including pitch, time, and spectral structure. The creation of language models stores the content of words, such as word frequency and n-gram probability, to improve speech recognition accuracy. During decoding, secret word strings are carefully analyzed to identify the statements most likely to be true. This combination will help increase the strength and accuracy of the entire authentication system.

Text-to-speech recognition uses a variety of techniques, including TTS synthesis, traditional use of pre-written human speech, and combining small words such as phonemes or diphones to form spoken words. Another traditional method is synthesis, which models the voice and makes speech by controlling the way it is spoken. In contrast, articulatory synthesis is an advanced method that simulates the movement of internal organs such as the tongue, lips, and jaw to produce speech. Each technology has its own unique qualities and instincts for human-like communication.

2. Towards Unsupervised Speech-to-text translations

The main step of the framework involves initializing a speech-to-text (ST) system using bilingual dictionaries derived from a language corpus. The dictionary intelligently maps each part of speech associated with a word to the target dictionary. The system seamlessly translates each part of the speech word for word in the language, without seeing the speech. This recommendation guarantees the accuracy and content of the translation for various languages.

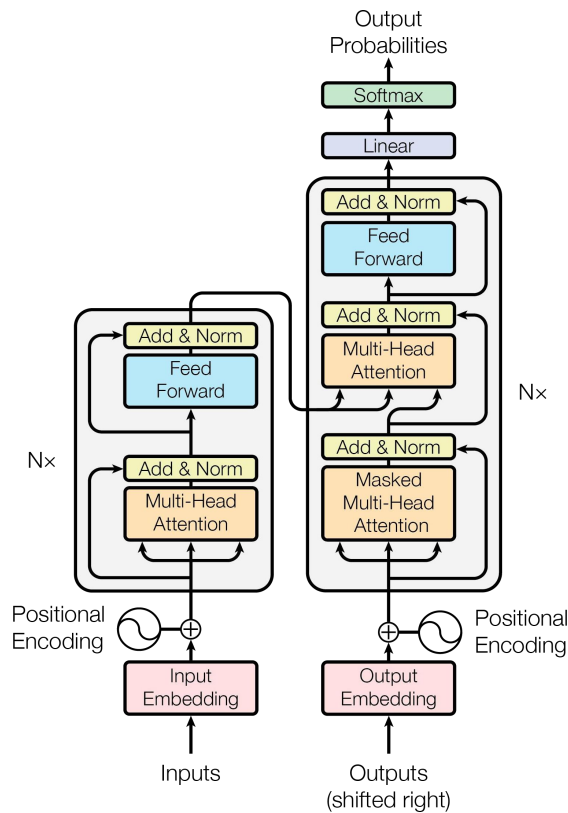
Word-by-word translation uses the unsupervised bilingual dictionary generation (BDI) algorithm to obtain a cross-word mapping from the source embedding source to the target embedding source. To improve the word-to-word translation process, a language model (LM) is integrated into the decision-making phase of context-aware search. To emphasize grammatical accuracy and improve translation, a level denoising autoencoder (DAE) is used to regularize the output.

During training, the Speech2Vec model and source text were created using Word2Vec and fastText; Both were configured to use default settings and without subword instructions, enabling 100-dimensional embedding of speech and text. The use of VecMap and MUSE follows the same principle as their original authors. Use KenLM for language modeling and follow its default settings to represent LM as a 5-gram number. Finally, the DAE is used as a 6-layer Transformer with specific dimensions, including embedding and hidden layer sizes of 512, feedforward sublayer size of 2,048, and 8 listener heads. This generalization ensures the efficiency, accuracy of the translation.

3. Attention is all you need [1]

This article introduces Transformer, a new network based on monitoring processes, eliminating the need for duplication and coordination. The trace function in this illustration shows the query and method of key value for output, where the query, key, value, and output are all represented as vectors. The output is given by the weight of the key; Each weight is determined by a relational function that compares the query to the corresponding key.

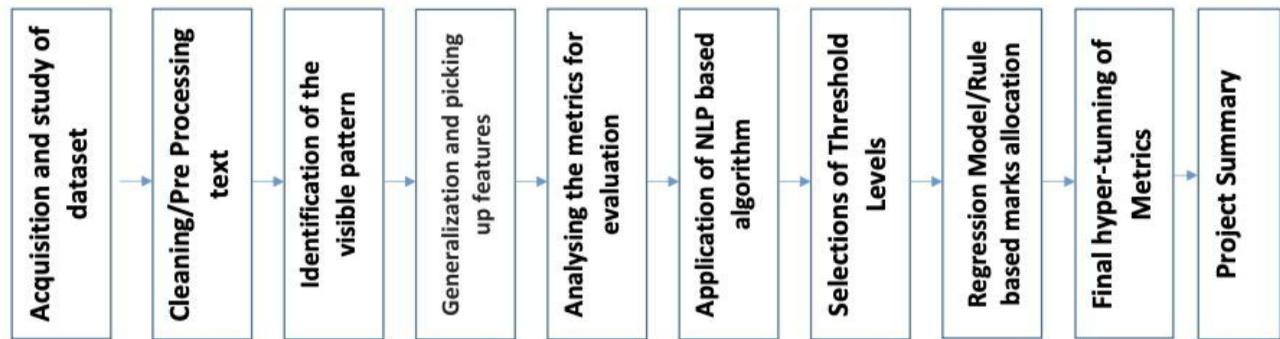
In this architecture, the encoder converts the input sequence into a series of continuous representations. The decoder then uses these representations to create the values of the symbols. The model follows an autoregressive approach that takes pre-generated characters as additional input at each step of the generation process. All of Transformer's models include a self-monitoring process that provides a powerful framework for capturing relationships in data.



Like other transformation models, this paper adopts learning embeddings to transform input tokens and output tokens into vectors with d_{model} dimensions. This technique involves transforming the output of the decoder into the predicted probability of the next symbol in the sequence using a linear transformation model and a softmax function. This ensures that model effectively processes input and output tokens, helping to generate accurate predictions for subsequent tokens in the sequence.

4. NLP-based Automatic Answer Evaluation

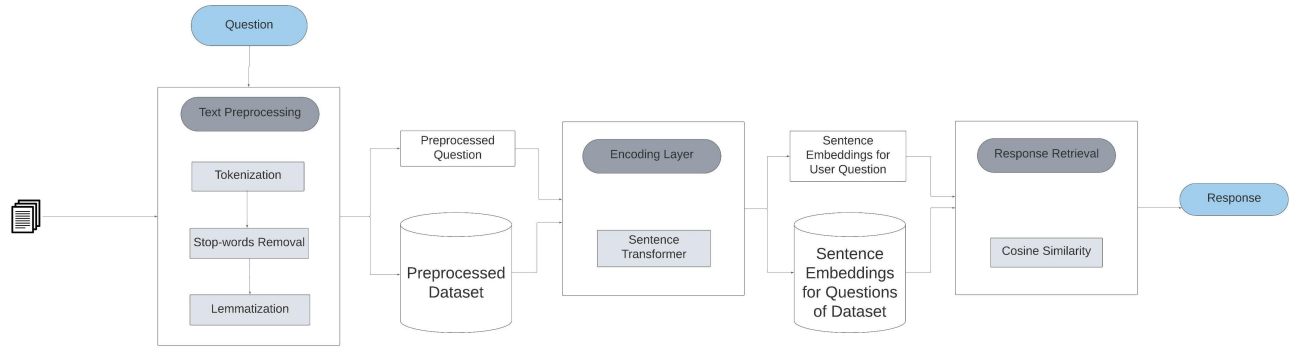
This study shows how to process words effectively by analyzing responses using algorithmic methods. The scoring process involves extracting text from answers, comparing the extracted content to previously stored correct answers to calculate consistency, and assigning a weighted value to each metric. To analyze the data, the research uses themes based on the data collected to create a summary from the data collected.



The study initiates by acquiring and examining the dataset, followed by a meticulous cleaning process and concluding with data preprocessing. Text preprocessing encompasses tasks such as tokenization, stopwords removal, lemmatization, and duplicate word elimination. To evaluate various similarities, a word dictionary incorporating frequency and bigram information becomes a pivotal component of the algorithm. This foundational function is instrumental in both generating summaries and quantifying the similarity index between any two provided documents. The 'sklearn' library, specifically its 'cosine_similarity' module, is employed to measure similarity.

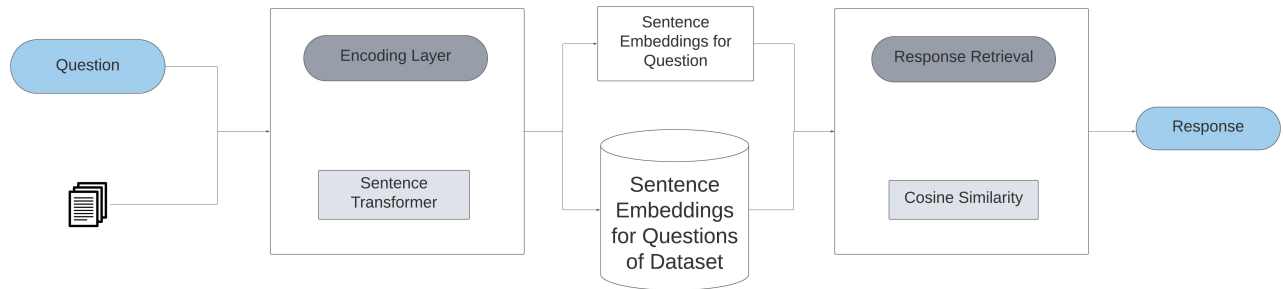
5. A Novel Approach for Building Domain-Specific Chatbots by Exploring Sentence Transformers-based Encoding [10]

In this study, we investigate the efficacy of two distinct models: the Term Frequency-Inverse Document Frequency (TF-IDF) and the Sentence Transformers model. Sentence transformers specialize in encoding sentence meaning and capturing semantic similarity, facilitating the generation of responses that are contextually coherent and relevant. The pre-processing phase in this paper relies on the Natural Language Toolkit (NLTK), a widely acclaimed machine-learning technique, incorporating tasks such as tokenization, stopwords removal, and lemmatization. The Encoding Layer section offers two approaches to encode or transform input sentences into their vector representations, enhancing the model's comprehension of the input data.



$$TF\text{-}IDF(t) = TF(t) \circ \theta IDF(t)$$

TF-IDF (Term Frequency-Inverse Document Frequency) is a technique employed for data retrieval relevant to user queries. This method assigns a TF-IDF value to terms, emphasizing those that are important yet frequent. High Term Frequency (TF) and low Inverse Document Frequency (IDF) values contribute to a significant TF-IDF value, ensuring the weight of such terms is retained during data retrieval.



On the other hand, Sentence Embeddings involve converting human language sentences into vectors of numbers. Using a transformer model, input sentences are processed to extract features for words, creating a set of consistent features for every word. These features serve as new axes or dimensions, and the model assigns scores to these identified features, forming a vector of numbers corresponding to the nature of each word. The all-MiniLM-L6-v2 model is employed for sentence embeddings, designed specifically for this purpose. The transformer layers within the model play a key role in capturing semantic meaning and contextual information in sentences. Finally, mean pooling is applied to the vectors obtained from the transformer layers, generating a fixed-length vector

Cosine similarity serves as a straightforward measure of similarity between two vectors. Following the encoding layer, the next step involves answer retrieval, and in this context, cosine similarity is employed.

$$\text{Cosine Similarity}(V_1, V_2) = \frac{V_1 \circ V_2}{|V_1| \circ |V_2|}$$

6. Sentence Similarity Based on Semantic Vector Model [8]

This article is dedicated to directly calculating the similarity between short texts (especially sentences). The algorithm takes into account semantic, structural and word order information in the sentence. The algorithm uses data from a database called How-net to calculate the semantic similarity between two sentences. The use of database content leads to ways to integrate human knowledge. In CNKI, each word is represented as a collection of meanings, Units are the smallest units with hierarchical relationships and form a tree structure. This algorithm provides an overall measure of semantic similarity between sentences by calculating semantic similarity based on the distance between semantic similarity in the semantic distribution.

Simword(q(i),w(j)) is a word similarity of q(i) and w(j).

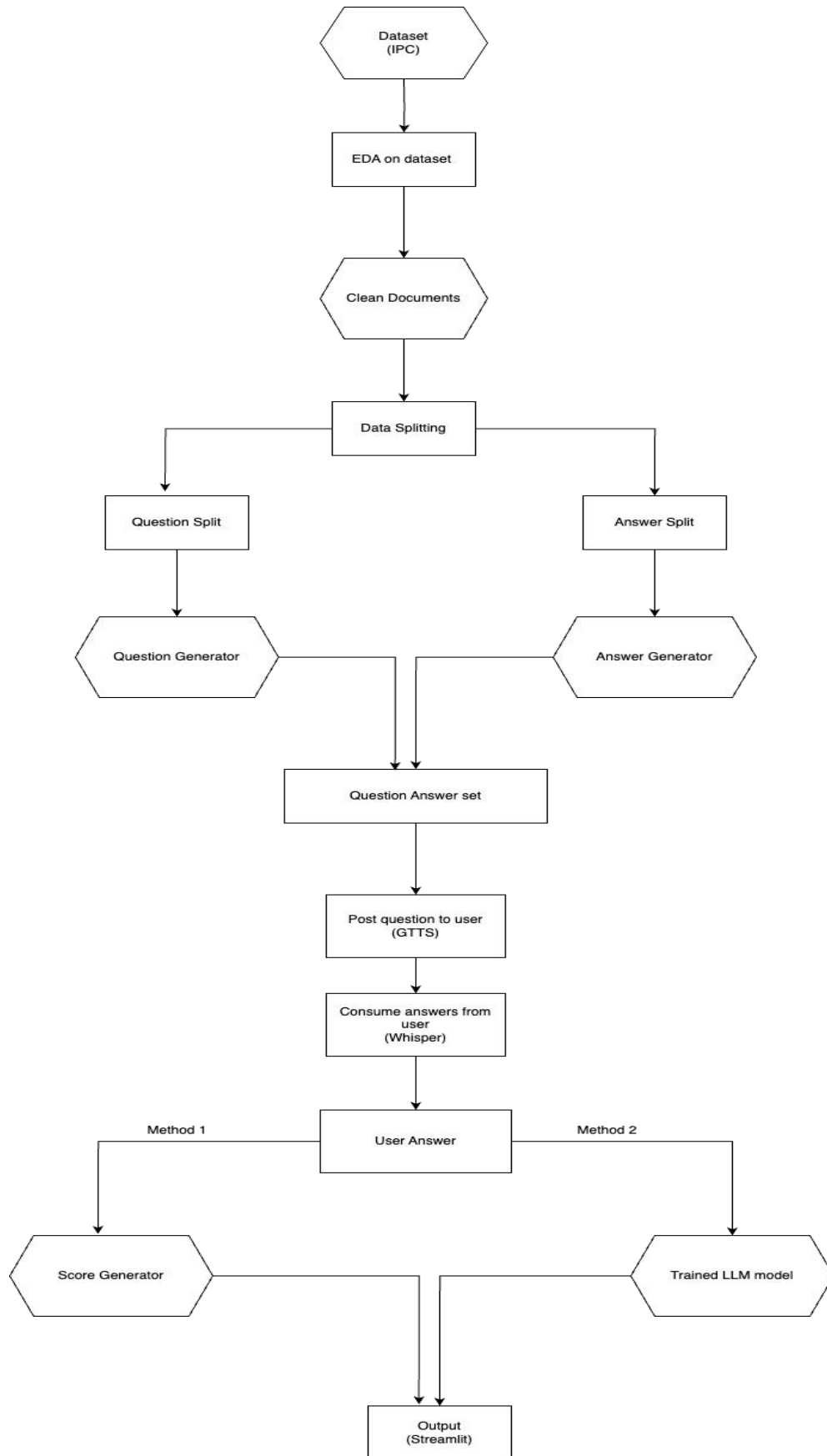
$$s_i = \begin{cases} 1, & q_i = w_j, 0 \leq j \leq n \\ \max\{sim_j\}, & sim_j = Sim_{word}(q_i, w_j), 0 \leq j \leq n, \max\{sim_j\} > \mu \\ 0, & \max\{sim_j\} < \mu \end{cases}$$

By adjusting the semantic similarity, semantic vector between the sentences can be defined as the cosine coefficient between the two semantic vectors:

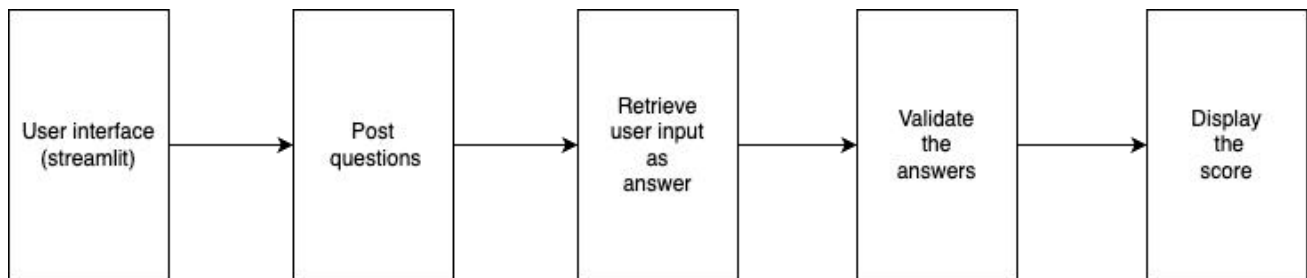
$$Sim_s = \frac{S_1 * S_2}{|S_1| * |S_2|}$$

The similarity method discussed in this paper introduces two parameters that require determination before implementation: a threshold for deriving semantic vectors and a factor 'e' for weighting the significance between semantic information and word order information. Through empirical exploration, the study found that a semantic threshold of 0.2 and a factor 'e' value of 0.85 yielded optimal results. Notably, the method that combines semantic and word order information was identified as the second-best approach, showcasing its effectiveness. In contrast, the method solely based on word order exhibited poor outcomes, suggesting that word order plays a subordinate role in determining sentence meaning compared to the combined influence of semantic and structural information.

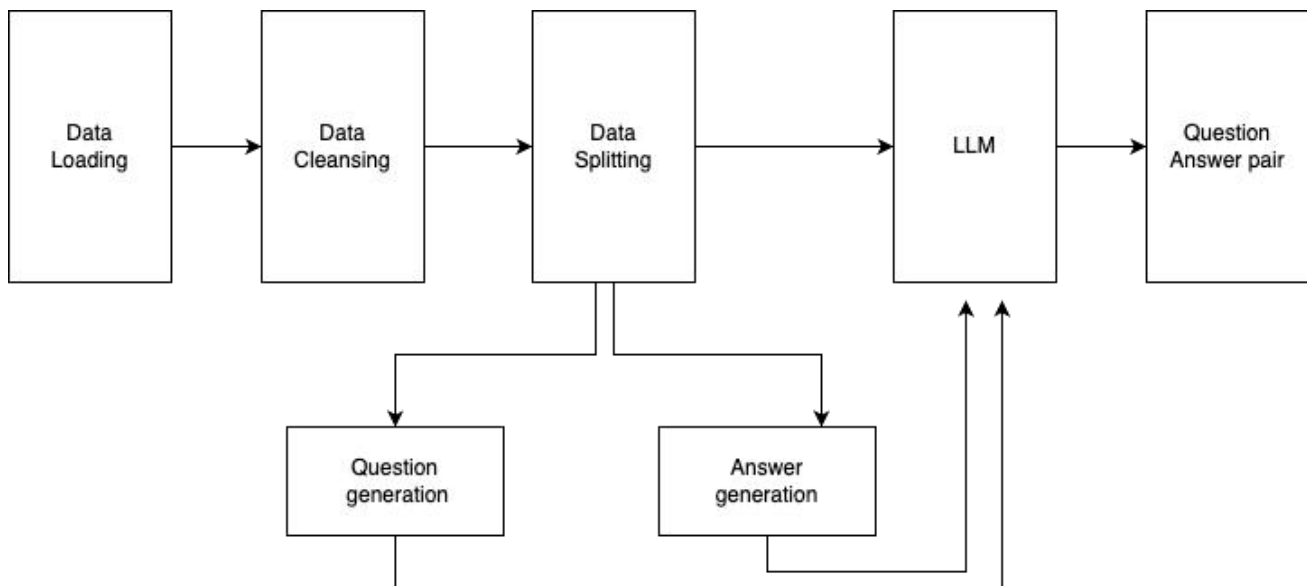
ARCHITECTURE



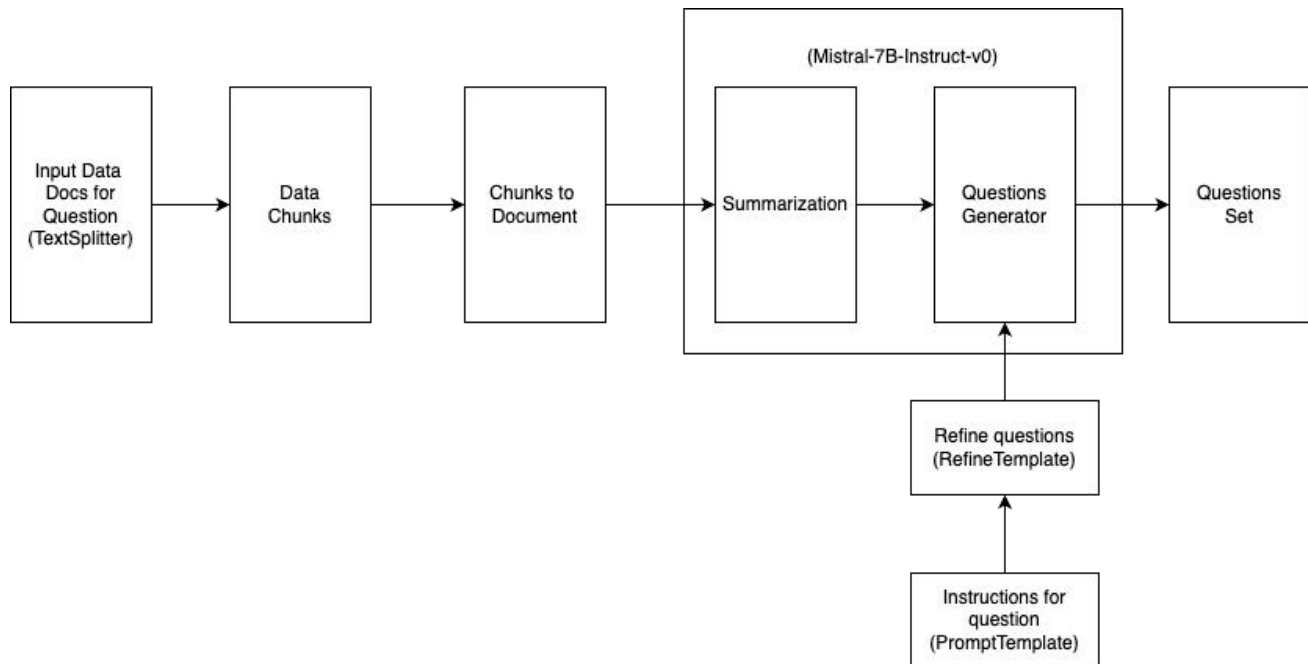
User level data flow



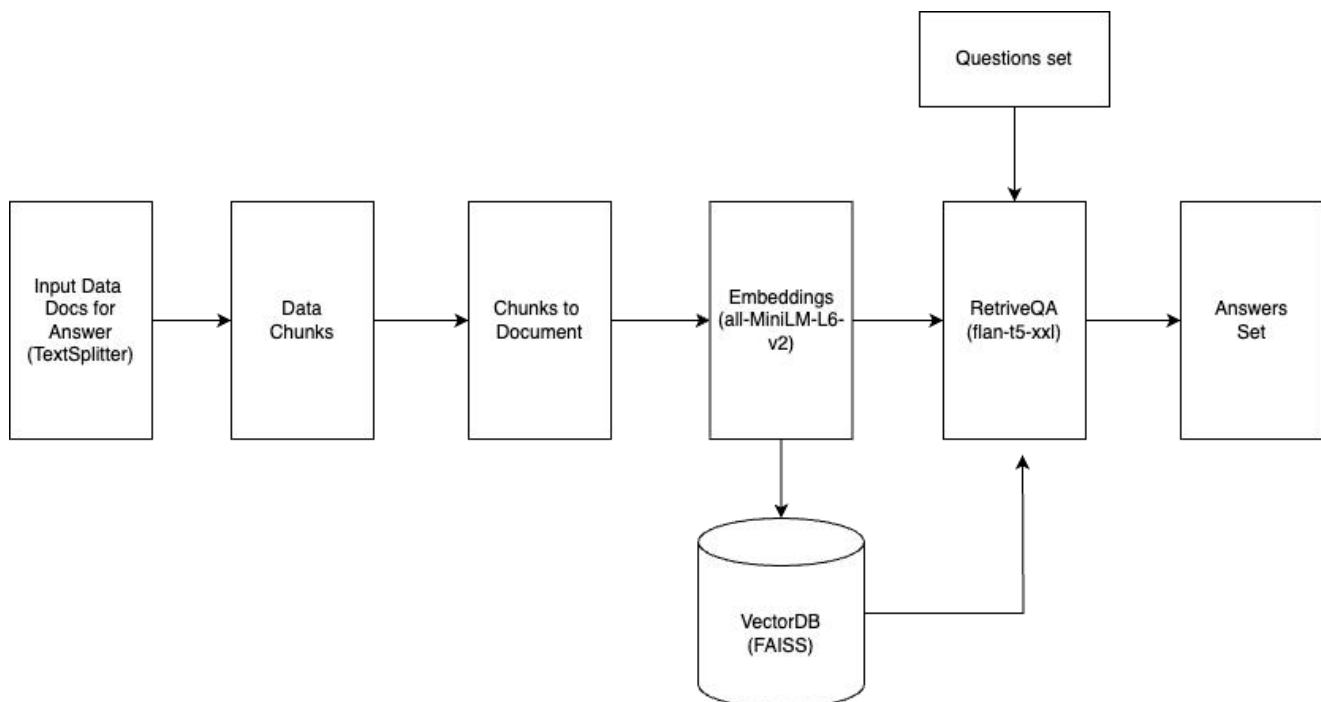
Data pre-processing



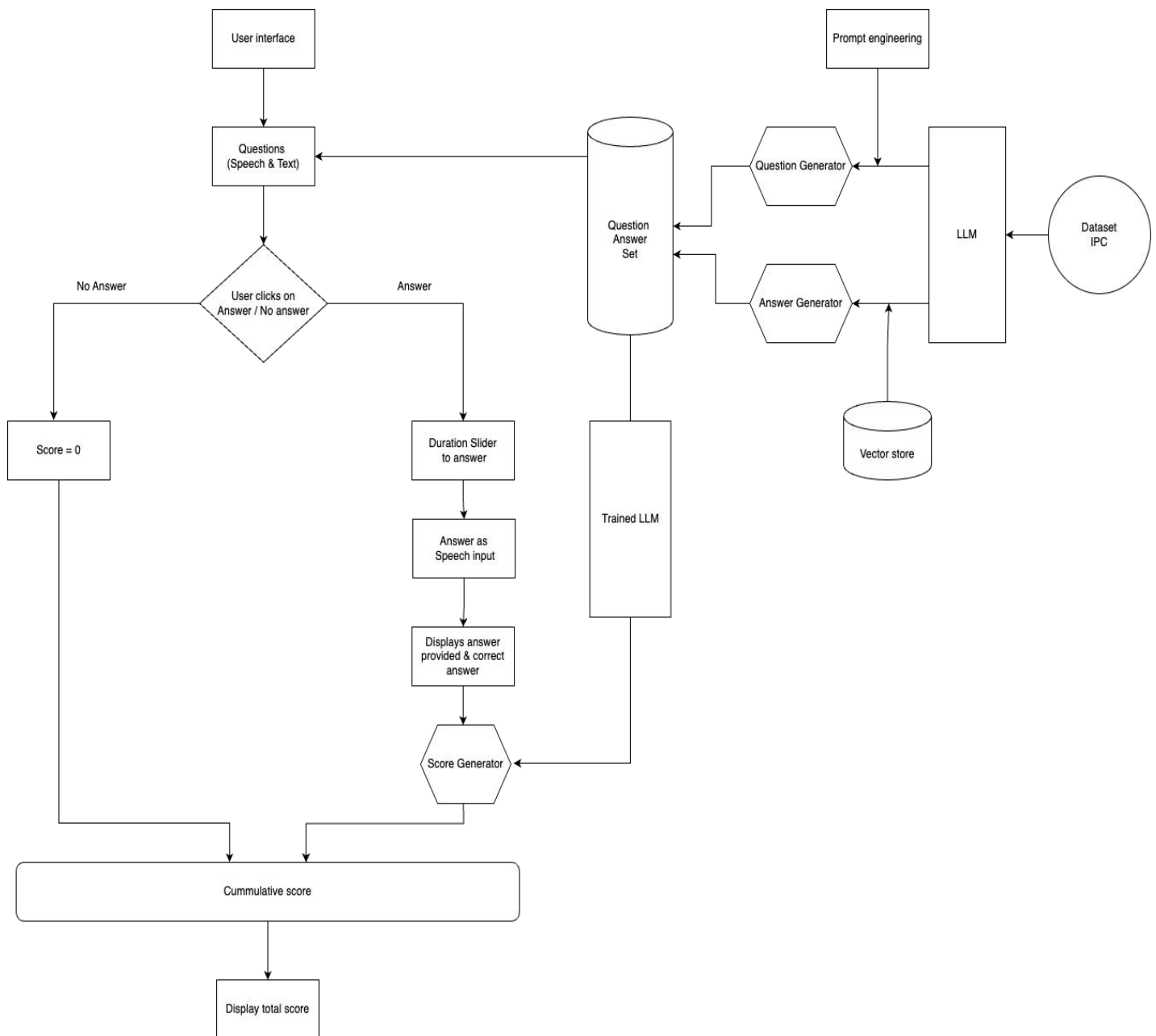
Question Generation Flow



Answer Generation Flow



Business Process Flow



DATA PRE-PROCESSING

PyPDF2

PyPDF2 is a multi-purpose Python library designed for processing PDF files, providing various functionality to read, edit and extract information from PDF files. Using PyPDF2, developers can easily merge, split and rotate PDF pages, as well as extract text and images. This library provides a user-friendly interface for working with a variety of PDF files, simplifying complex tasks associated with PDF management.

One of the main advantages of PyPDF2 is its simplicity and ease of integration into Python applications. Whether you need efficient processing of files or extraction of specific files from PDFs, PyPDF2's comprehensive toolset allows developers to process PDF files because it is useful in many aspects, such as data extraction, data management, and reporting tools.

PdfReader

PdfReader is a key part of the PyPDF2 library and provides developers with powerful tools to read and extract information from PDF files. This reader class provides capabilities for data extraction, analysis, and analysis control by allowing Python applications to access the content, structure, and metadata of PDF files. With PdfReader, developers can browse pages, extract text and images, and store important information in PDF files.

Its intuitive interface and powerful features make it easy to work from text extraction to advanced operations and allow developers to create interactive applications. Import and export PDF content. The PdfReader class is especially useful for tasks that require efficient access to PDF files, making it the core of the PyPDF2 library's PDF file management tools: Python.

RE

The 're' library in Python is a powerful tool that provides support for regular expressions, allowing developers to compare and manipulate data. A regular expression, often abbreviated as regex or regexp, is a string of characters that defines a search pattern. Through this library, developers can create and use these models to search, match, and manage strings efficiently. This library is an essential tool for tasks such as text parsing, validation, and extracting custom patterns from strings, making it an essential part of Python programming.

Data cleansing

Data cleansing is the process of identifying and correcting errors or inconsistencies in a dataset to improve its quality and accuracy. Important steps in preparing this data include handling missing values, removing duplicates, and correcting errors. Good data management will help you better understand and derive value from your data.

Data is extracted from a PDF file, and a regex pattern is applied to each page to transform it into a structured document. The transformation includes capturing section identifiers, section names, and comments from the content of the PDF. This process enables the creation of a well-organized document with distinct section IDs, corresponding section names, and associated comments, facilitating further analysis and interpretation of the extracted information. In our case each document consist of section_id, section_name and comment or case study related to sections and subsections.

RecursiveCharacterTextSplitter

"TextSplitter" typically refers to a utility or class designed to facilitate the process of dividing a given text into smaller, more manageable segments based on certain criteria. This functionality is especially useful in natural language processing, data preprocessing, or any task that involves breaking down large text datasets into meaningful chunks. A TextSplitter might employ various techniques, such as pattern matching, regular expressions, or specific delimiters, to achieve accurate and efficient segmentation.

This text splitter is the recommended one for generic text. It is parameterized by a list of characters. It tries to split on them in order until the chunks are small enough. The default list is ["\n\n", "\n", " ", ""]. This has the effect of trying to keep all paragraphs (and then sentences, and then words) together as long as possible, as those would generically seem to be the strongest semantically related pieces of text.

OS

The os module in Python is a useful tool that provides a platform-independent interface to interact with functions. It allows Python programs to manipulate various files and directories, manage processes, and access special functions. This module contains functions useful for the connection between Python scripts and the operating system and executing shell commands. It plays an important role in creating powerful and portable applications by removing most of the complexities related to data and operations, making it an integral part of the Python standard library.

Document creation

List of documents is created considering chunks of huge dataset loaded, we have adopted a multi-step approach to create a list of documents from a large dataset. The process involves scanning through the dataset by loading a directory. The first step is to identify a regex pattern to extract section numbers and their associated titles. Subsequently, we extract comments that contain subsections and case studies related to each section. Once a set of section data is gathered, we create a document that includes the section ID, section name, and associated case studies.

Our dataset comprises approximately 511 sections, numerous subsections, and corresponding case studies. To facilitate seamless embedding, we leverage Langchain document loaders in the document creation process. We have used Langchain document loaders in order to create the documents so that embedding becomes seamless.

Embeddings

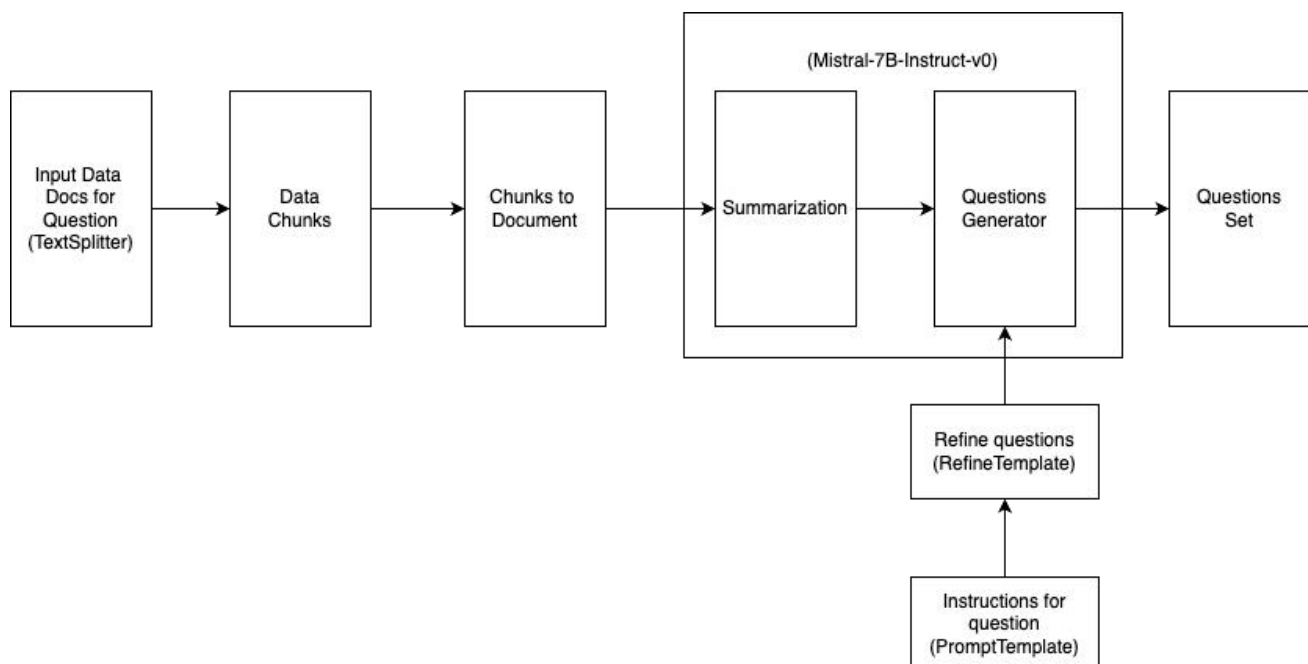
Word and sentence embeddings are techniques used in natural language processing (NLP) to represent words or sentences as a continuous vector space. Word embedding aims to capture the semantic relationship between words by drawing them into vectors in a high-dimensional space. Methods such as Word2Vec, GloVe, and FastText are commonly used to create word embeddings. In these placements, words with similar content are kept close to each other in the vector space, allowing the model to capture similar content and similar words.

On the other hand, sentence embedding extends this concept to represent the entire sentence as a vector. Unlike bag-of-words or TF-IDF representations, sentence embeddings aim to encapsulate the meaning of a complete sentence in a continuous vector. Techniques such as Universal Sentence Encoder, InferSent, and BERT are used to generate sentence embeddings. These embeddings help better understand sentence semantics, facilitating applications such as sentiment analysis, data matching, and machine translation. Word and sentence embeddings play an important role in supporting NLP models, making them easier to understand and process better.

QUESTION GENERATION

In the question generation workflow, the initial step centers around breaking down a lengthy text into smaller, digestible portions or document chunks. Each of these chunks is then subjected to a summarization process guided by a predefined PromptTemplate. This template not only assists in condensing the information within each segment but also aids in the generation of relevant and targeted questions tailored to the content of that specific chunk. The base model that is employed here is Mistral-7B-Instruct-v0.1. By employing this method, the intricate details and nuances present in the original text are distilled into concise summaries, laying the foundation for the subsequent question-generation phase.

The second stage involves leveraging the extracted questions from individual document chunks to construct a more expansive and comprehensive list of inquiries. This aggregation process extends and combines the questions generated from each segment, resulting in a holistic set of queries that collectively cover a broader scope of the entire text. The approach not only enhances the granularity of understanding within each document chunk but also ensures that the amalgamated list captures the diversity and intricacies inherent in the larger context of the original text. Overall, this method provides a systematic and effective way to distill insights from extensive textual content through targeted summarization and question generation.



PROMPT TEMPLATE

A prompt template serves as a structured framework for generating specific types of content or responses from language models. Existing models are models designed to generate specific content or responses from language models. It often contains placeholders or directives that direct the language structure to create the desired object. This model is important for fine-tuning and adapting the response model to a specific use or application. The current model provides a way to download the same data and targets by providing a pre-designed model, making them useful in language processing, content creation, and discussion.

The effectiveness of current models lies in their ability to pick up language patterns to produce coherent responses and content. Whether used for Q&A, content creation, or other language applications, well-designed templates increase the accuracy and reliability of the language produced. Designing this style often requires a balance between specificity and flexibility, allowing users to customize the content produced while adapting to a variety of ideas and activities.

Mistral-7B-Instruct-v0.1

Mistral-7B-Instruct-v0.1 Large Language Model (LLM) is a fine-tuned training variant of the Mistral-7B-v0.1 built-in model that incorporates information from various discussions in the public record. Mistral 7B uses the Sliding Window Attention (SWA) mechanism (Child et al., Beltagy et al.) to improve the performance of each layer by allowing it to focus on 4,096 previously hidden states. The main improvement is the operating cost of $O(\text{sliding_window.seq_len})$. The updated strategy for FlashAttention and xFormers provides a significant 2x speedup for 16k sizes and 4k windows. Special thanks to Tri Dao and Daniel Haziza for their valuable collaboration that allowed us to quickly integrate these developments in a short time.

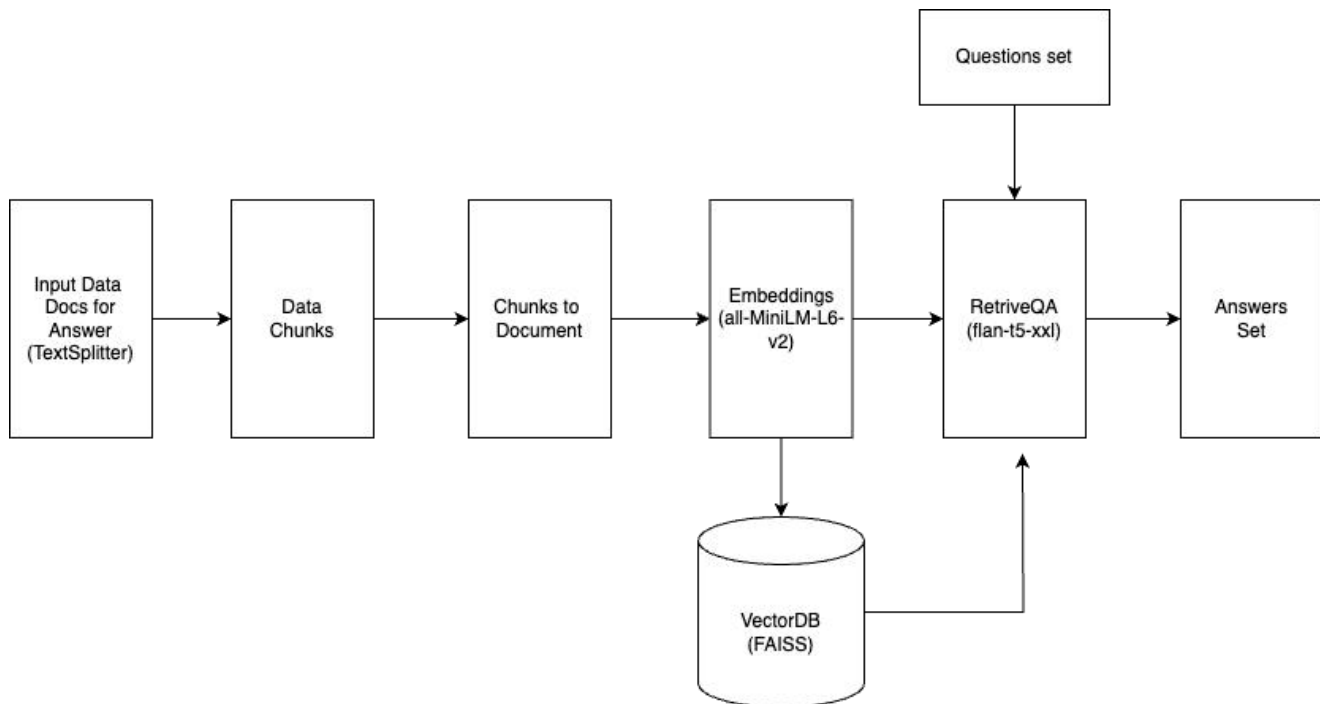
The sliding window view takes advantage of Transformer's hierarchical structure to expand the window size set in another area. For example, the token at position i in layer k will target the token $[i - \text{sliding_window}, i]$ in layer $k-1$. This allows advanced tokens to access information from a more distant past than was initially visible through colored models. This approach ensures that the model has the ability to capture many elements and dependencies in real-world applications.

GGUF is a new format introduced by the llama.cpp team on August 21st 2023. It is a replacement for GGML, which is no longer supported by llama.cpp.

ANSWER GENERATION

The process of generating answers involves breaking down the input text into smaller document chunks, each of which is summarized, and embeddings are created to represent them effectively. These embeddings are then stored in a vector database. By using the Google Flan model, the system retrieves the most similar document when a question is posed, enabling the generation of precise answers. This iterative approach ensures that each question is systematically processed through the summarized document chunks, and relevant answer is extracted, providing accurate and contextually appropriate answers.

This methodology demonstrates a systematic and effective way to handle a variety of questions with the help of document chunking, summarization, and vector embeddings. The integration of the Google Flan model enhances the retrieval process, ensuring that the system can efficiently pinpoint the most relevant information in response to user queries. The step-by-step iteration through each question ensures a thorough exploration of the underlying content, contributing to the overall effectiveness of the answer generation process.



Sentence Transformer

Sentence Transformer is a versatile library designed to convert sentences into stable formats suitable for multi-language processing (NLP). Developed by UKPLab at the Technical University of Darmstadt, the library uses a pre-trained Transformer model to encode sentences for a dense representation of images. Sentence Converter is unique in its ability to create useful content that supports applications such as similarity searching, category searching, and data retrieval. The library supports a variety of Transformer architectures and provides users with intuitive tools to experiment with sentence embeddings, making it useful for researchers and developers working on a variety of topics.

One of the important features of Transformer is that it supports many types of pre-learning, including models that have learned more than one language, allowing users to study and work with more than one language. The flexible and efficient library makes it the first choice for tasks such as data storage, query answering and similar semantic tags, contributing to the increasing level of NLP applications by combining the technology of Transformer-based technologies.

all-MiniLM-L6v2

all-MiniLM-L6-v2 represents a sentence-transformers model, proficient in mapping sentences and paragraphs to a 384-dimensional dense vector space, making it applicable for tasks such as clustering or semantic search. The project's core objective involves training sentence embedding models on extensive sentence-level datasets, employing a self-supervised contrastive learning objective. all-MiniLM-L6-v2 model is pretrained on top of nreimers/MiniLM-L6-H384-uncased model, it was fine-tuned on a vast dataset of 1 billion sentence pairs. The training involves a contrastive learning objective, wherein the model predicts which among a set of randomly sampled sentences was paired with a given sentence in our dataset.

The implementation of the all-MiniLM model involves the use of a compact and powerful language model designed for many operating languages. all-MiniLM models are known for their efficiency and effectiveness in capturing textual representations. The term “all-MiniLM” indicates general use of the MiniLM architecture and possibly indicates a way to use the model for many tasks in different contexts.

VECTORISATION

Vectorization in natural language processing (NLP) involves transforming data into numerical vectors, allowing machine learning algorithms to process and analyze language. This process is important for connecting the differences between the symbols of words and the numerical representation required by machine learning models. Vectorization uses a variety of techniques such as Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and more recently word embeddings such as Word2Vec, GloVe, and FastText. In particular, word embeddings capture semantic relationships by representing words as dense vectors in a continuous vector space. Vectorization plays an important role in enabling algorithms to understand concepts and different concepts in textual data, ultimately improving the performance of NLP applications, including analysis theory, classification and interpretation of texts.

Vector Store

In the context of data storage and retrieval, vector storage generally refers to systems that store data in a vectorized or digital format, allowing similar searches and retrieval of the vector representation. This approach is often used in data retrieval, machine learning, and word processing. In vector storage, data entities (such as data or objects) are represented as vectors in high-dimensional space, where similarity of vectors corresponds to similarity or dissimilarity. Be good in the following places.

Vector storage is especially useful in situations where search similarity and repetition play an important role, such as recommendations, search engines, and recommendations. These systems use the mathematical properties of vector spaces to quickly identify and store query-like objects or data. Vectors capture important features or characteristics of data, enabling comparison and efficient search. Create meaningful representations in vector stores using a variety of techniques such as word embedding, data embedding, and neural network-based representation.

Using vector storage involves indexing and adjusting vectors to support fast and accurate searches. Popular libraries and frameworks such as Faiss, Annoy, and Elasticsearch provide tools for creating and querying vector stores. The use of vector storage is becoming more common in today's data processing applications and offers a powerful way to enhance search and visualization.

FAISS

Facebook AI Similarity Search, is an open source library developed by Facebook for efficient similarity search and high vector space integration of large datasets. Particularly popular in the machine learning community, Fais is designed to handle large vectorized data; This makes it ideal for tasks such as nearest neighbor search, effective access consistency, and unity.

One of Faiss's greatest strengths is its use of advanced techniques (such as data transfer and product measurement) to search for peers. The library supports both CPU and GPU implementation, making it fast and efficient. It is widely used in applications where similar investigations are important, including power, consensus processes, images and fallbacks, and data separation. Focusing on speed and efficiency, Fais has become the tool of choice for doctors working with large vector files, providing a solid foundation for producing similar reference books in different formats.

FAISS-CPU

The "faiss-cpu" component is a special version of Fais designed to run on a central processing unit (CPU). Although faiss still supports the use of GPU for faster processing, the need for "faiss-cpu" arises when GPU capacity is not available or when computational requirements can only be met by CPU resources.

Using "faiss-cpu" cpu" allows developers to leverage the power of Fais on machines without work pressure, interfacing with a variety of hardware. This change is important for applications and environments where GPU resources are used. They may be limited or unavailable. Even with cloud-based solutions. Different computers.

Vector spaces

In natural language processing (NLP), vector spaces are simple mathematical structures used to represent words, sentences, or data as mathematical vectors. In this space, each word or phrase is assigned a unique vector, allowing mathematics to capture these relationships. In the vector space model, words with similar content are placed close to each other so that the algorithm understands the content and communication. Technologies such as Word2Vec, GloVe, and FastText create word embeddings that place a word in a fixed space vector. Similarly, sentence embeddings extend this idea to represent the entire sentence, improving the content and understanding of language in NLP, such as emotional analysis, good information compatibility, and machine translation. The use of vector spaces makes it possible to compare the content of words in a way that has a good semantic relationship.

Hugging Face Hub

Hugging Face Hub is a powerful platform that serves as the foundation for positive language processing (NLP) models, information, and other resources. It provides a collaborative environment where developers, researchers, and practitioners can share, explore, and collaborate on modeling and knowledge in NLP. Users can easily access pre-trained models, try variations, and contribute their own models to support community collaboration.

HuggingFace Hub's user-friendly interface simplifies model deployment, allowing developers to easily integrate NLP capabilities into their applications. The center hosts many models and datasets and enables innovation in NLP by encouraging collaboration and knowledge sharing. The platform has become an integral part of the NLP ecosystem, allowing developers worldwide to benefit from and contribute to new advances in natural language processing.

RAG

Retrieval Augmentation Generation is a method for increasing the knowledge of a large language model (LLM) by incorporating additional information. Although LL.M.s can describe a variety of disciplines, their understanding is limited to information published before a particular teaching period. The development of innovation involves the process of comparing basic elements with pattern demonstrations in order to expand the ability to take into account special information or post-release information.

LangChain provides a variety of features designed to assist in the development of applications for query answering and, more specifically, applications that use retrieval augmented rendering (RAG) methods. These components help LL.M.s hone their skills by enabling them to synthesize and construct responses based on more specific information from original educational materials, thereby contributing to the development of advanced applications and context-aware artificial intelligence.

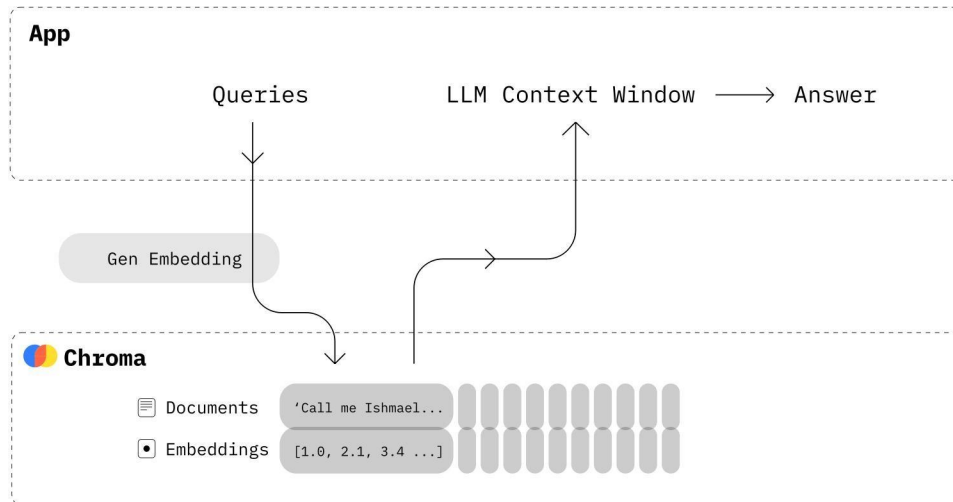
A standard Retrieval Augmented Generation (RAG) application comprises two primary components:

Indexing: This involves a data ingestion pipeline that sources information from a designated origin and indexes it. Typically, this process occurs offline, enabling efficient organization and storage of the data for subsequent retrieval.

Retrieval and Generation: The core of the RAG application, this component operates in real-time during user interactions. When a user submits a query, the chain dynamically retrieves pertinent data from the index and feeds it into the model for processing. This seamless integration ensures that the user receives accurate and contextually relevant responses based on the information retrieved from the indexed data.

ChromaDB

Chroma is an open-source vector database designed to efficiently manage and process vectorized data, especially in the field of hybrid data and similar systems. ChromaDB is designed to simplify the storage and retrieval of vectorized data, especially for relational and content-intensive applications. The framework helps create and manage data containing vector embeddings, connecting with linguistic processing (NLP) and other machine learning methods.



ChromaDB specializes in providing a scalable and robust solution that delivers vectorized data that is efficient, repeatable and robust. Its architecture is designed to accommodate various types of vector embeddings; This makes it suitable for a variety of applications, from data visualization to image recognition. ChromaDB is designed with extensibility in mind and supports easy integration with popular vectorization technologies and libraries, making it an essential tool for researchers and developers alike. It is an effective solution for managing vectorized data in complex applications.

MODEL TRAINING

Training or fine-tuning the Transformers models often depends on the availability of equipment and specific operations. These two methods require an in-depth understanding of how to enter data into the model and must carefully prepare the dataset. The most important thing to understand the various types of unemployment and their consistency with the characteristics of the dataset. This dual decision ensures an effective training procedure and allow the model to learn and adapt to the complexity of the material in a way that is beneficial to the intended task. SimpleTransformer model is being used to finetune the base model bert-base-cased with the question-answering dataset.

BERT (bert-base-cased)

BERT (Bidirectional Encoder Representation from Transformers) is a pre-learned language model known for its effectiveness in many NLP projects. A special variant called "bert-base-cased" is a version of BERT in which the difference between uppercase and lowercase letters is preserved. The model is well trained on large datasets and can be optimized for specific tasks such as document classification, domain recognition, and query answering.

"bert-base-cased" "The model has 110 million parameters and provides the representation of content embeddings. Its bidirectional architecture allows it to capture complex language patterns and bidirectional dependencies, making it well understood. It has good performance and Due to its ability to be used in many NLP applications, "bert-based-cased" is widely used by researchers and professionals as it provides training before the language structure is established.

The "bert-based cased" model is a pre trained language model designed specifically for English using Masked Language Modeling (MLM). It is presented in a special research form and is initially available from the relevant archives. More importantly, the model matters; such as recognizing the difference between lowercase and uppercase letters, the difference between "English" and "English". These features enhance the ability to capture subtle nuances and differences in the English language, providing greater representation in the context of deployment.

SimpleTransformer

Simple Transformer models are designed for specific language processing (NLP) applications and have unique features and capabilities that can be optimized for operational needs. The use of these models is often based on the same decision. First initialize the model function and then train the model using the `train_model()` function. Then the efficiency of the model is evaluated with the `eval_model()` function and predictions are made for anonymous data using the `Predict()` function. This design allows simple Transformers models to be used effectively in many NLP applications.

Question Answering module of SimpleTransformer aims to discover the answer to a posed question by considering both the question itself and the associated context. The anticipated answer may be represented as a text span extracted from the context or indicated as an empty string, denoting that the question cannot be answered based on the provided context.

The execution of Question Answering using Simple Transformers adheres to a standard sequence of steps. It commences by initializing a `QuestionAnsweringModel`, followed by training the model through the `train_model()` function. The model's performance is then assessed using the `eval_model()` function, and predictions on unlabeled data are generated with the `predict()` function. This systematic approach ensures the effective implementation of Question Answering tasks with Simple Transformers.

Dataset

The input data for Simple Transformers must be in the list of python dictionaries or JSON files containing such data. Each dictionary represents a different concept and its associated problems.

There are two terms in every dictionary: "context" and "qas".

context: contains the source of the problem statement.

qas: This is a list of questions and their answers following a specific format. The format of each dictionary in qas includes:

id: unique identifier (string) of the query to ensure consistency across all datasets.

question: a string representing the question.

is_impossible: Boolean value indicating whether the question given to the content can be answered.

answers: List of the right answers to the given questions.

Each answer in the list of python dictionary with the following properties:

text: A string containing the answer to the given question, which must be a string of elements.

answer_start: A number representing the starting index of answer in the context.

Training process

Simple Transformers library for training a Question Answering model based on the BERT architecture. The specified model type is "bert," and the specific variant used is "bert-base-cased." Key model parameters and training settings are defined through `model_args` and `train_args`. The training process is initiated with the `QuestionAnsweringModel` class from Simple Transformers, and the model is trained on a dataset named 'train' while evaluated on 'test' data. Noteworthy configurations include setting the training batch size, the number of training epochs, and enabling evaluation during training, with periodic evaluation steps.

The training settings include options such as reprocessing input data, overwriting output directories, and specifying the output and best model directories. The maximum sequence length for input data is set to 128 tokens, and training hyperparameters like batch sizes, the number of best predictions to consider (`n_best_size`), and the frequency of evaluation during training are defined. Additionally, the script integrates with the Weights & Biases (wandb) platform for experiment tracking, associating the project with "Question Answer Application" and assigning a name based on the selected BERT variant. The code exemplifies a structured approach to training a BERT-based Question Answering model using Simple Transformers, offering flexibility through customizable parameters and easy integration with external tracking tools.

WANDB

Weight and Bias (wandb) is a powerful experimental detection and visualization platform designed to simplify and improve machine learning. With wandb, scientists and data scientists can easily collect and compare experiments, see performance metrics, and collaborate with team members. The platform provides a central dashboard to organize experiments to easily search for training quality, hyperparameter tuning, and model evaluation. By integrating wandb into machine learning projects, users can gain better information about behavior and insights and performance patterns, allowing them to make more informed decisions throughout the development process.

Wandb's versatility goes beyond test tracking and extends to collaboration capabilities, allowing teams to share results, insights and insights in a collaborative environment. It supports many types of machine learning and integrates with popular libraries such as TensorFlow and PyTorch. The platform's user-friendly interface, combined with the ability to monitor and record experiments in real time, makes it useful for researchers and large-scale collaborative projects. Finally, wandb improves reproducibility, collaboration, and experimentation in the machine learning community.

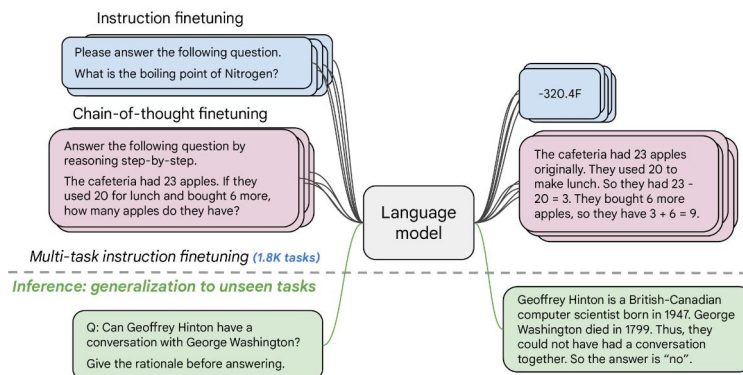
SCORE GENERATION

Two distinct methodologies have been employed for score generation in this context. The first approach involves conducting a similarity assessment by comparing user-provided answers with answers stored in the vector store utilizes cosine similarity to measure the likeness between the provided and stored answers, ultimately assigning a score based on the degree of similarity.

In the second approach, the learning model comes into play. In particular, the data generated from the question-answer system is used by models trained using the BERT model. The model evaluates the user's answers and provides a score up on the user's understanding of the appropriate models and the relationship between the answers in the answer. These two approaches provide flexibility and robustness for both vector-based similarity analysis and fuzzy understanding in language learning models.

FLAN-T5-XXL

FLAN-T5, introduced in the research paper "Scaling Instruction-Finetuned Language Models," represents an upgraded iteration of T5 that underwent fine-tuning across a diverse range of tasks. Specifically, FLAN-T5-xxl underwent fine-tuning on an extensive corpus of text data that lacked explicit content filtering and assessments for existing biases. Consequently, the model inherits a broader contextual understanding from its training data, offering enhanced capabilities for various language-related applications.



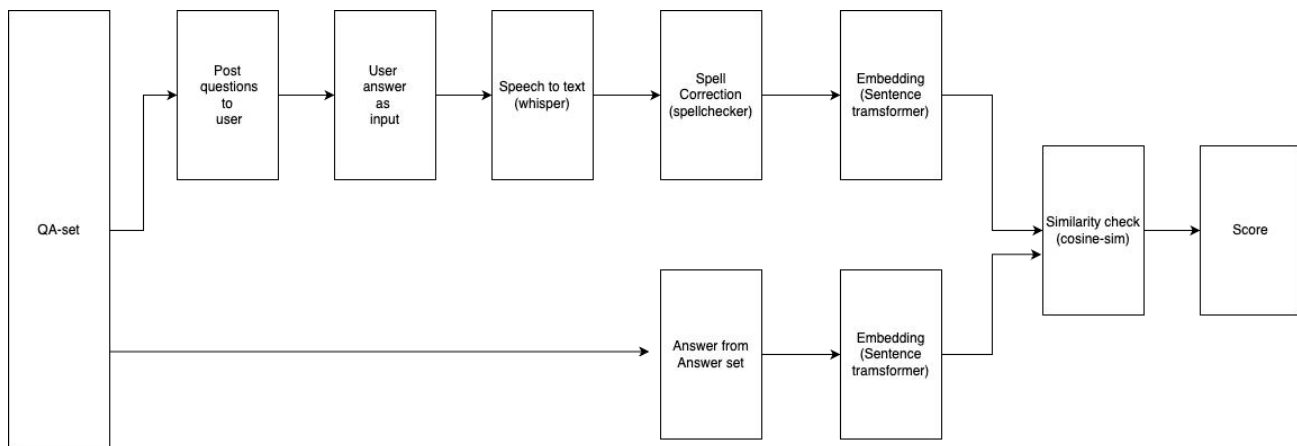
Cosine similarity

Cosine similarity is similarity a metric that is used to measure the similarity between two vectors in different spaces. Cosine similarity, which is especially common in natural language processing and data retrieval, indicates direction similarity by determining the co-sine of the angle between the two vectors.

In fact, when applied to data files represented as vectors, cosine similarity is useful for operations such as data comparison, data feedback, and recommendations. This metric, which analyzes the cosine of the angle of the vector representation of text, provides a reliable measure of their compatibility and is useful for many applications in technology studies and data recovery.

$$\text{Cosine Similarity}(V_1, V_2) = \frac{V_1 \cdot V_2}{|V_1| \cdot |V_2|}$$

Calculating Cosine similarity involves determining the cosine of the angle between vectors, resulting in values between -1 and 1. A cosine similarity of 1 indicates similarity; so -1 indicates the opposite. This parameter is especially useful in cases where the magnitude of the vector is less important than the direction of the vector.



USER INTERFACE

Streamlit

Streamlit is an open-source Python library specifically crafted for the effortless creation of web applications. It streamlines the intricate task of transforming data into interactive web applications, making the development process notably straightforward and accessible for users with minimal effort. With its intuitive and user-friendly API, Streamlit provides a combination of widgets, charts, and reports, making it accessible to data scientists and developers. Popular for its simplicity and speed, this tool provides a quick way to model and illustrate data-driven visualization without the need for web skills.

Speech input - sounddevice

Python's 'sounddevice' library is a versatile and easy-to-use tool for managing real-time audio input/output. With simplicity and performance as its main goal, sounddevice helps capture and play music directly from Python scripts. It supports many platforms, making it compatible with many operating systems. This library provides advanced communications for interacting with audio devices, allowing users to easily record audio input, play audio output, and play runtime music. sounddevice's API provides simple functions such as adjusting sample rates, adjusting channels, and managing sizes. It also supports different backends, including PortAudio and ASIO, providing flexibility to users with different needs.

Pydub

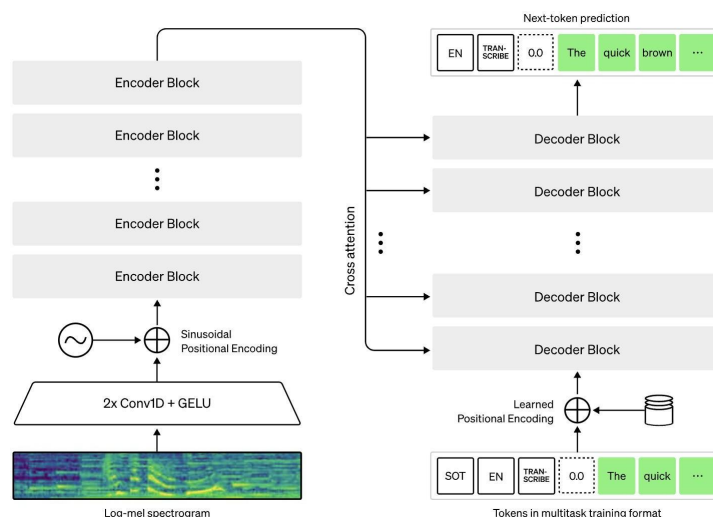
Pydub is a Python library that makes it easy to manage audio files through an easy-to-use interface. Pydub is built on top of FFmpeg and other audio tools, providing a variety of functions such as reading and writing audio files, converting between different formats, using effects, and performing simple operations such as slicing and merging.

One of the most important features of Pydub is the ability to manage multiple audio files; This allows users to focus on desired tasks without having to include difficult data points. The library supports various audio formats such as MP3, WAV and FLAC. Whether you want to extract tracks from audio files, apply filters, or convert between formats, Pydub is a powerful and versatile tool that makes things easy and efficient. Work as an expert Audio in the Python programming environment.

Speech to Text

Speech-to-text (STT) technology is a revolutionary branch of natural language processing that plays a vital role in converting speech into text. The STT system uses advanced algorithms and machine learning models to better understand and record speech. These systems use acoustic patterns to interpret sound signals, language patterns to identify language patterns, and language patterns to enhance situational understanding. This project has opted to make use of the Whisper API from OpenAI in order to convert speech to text more effectively.

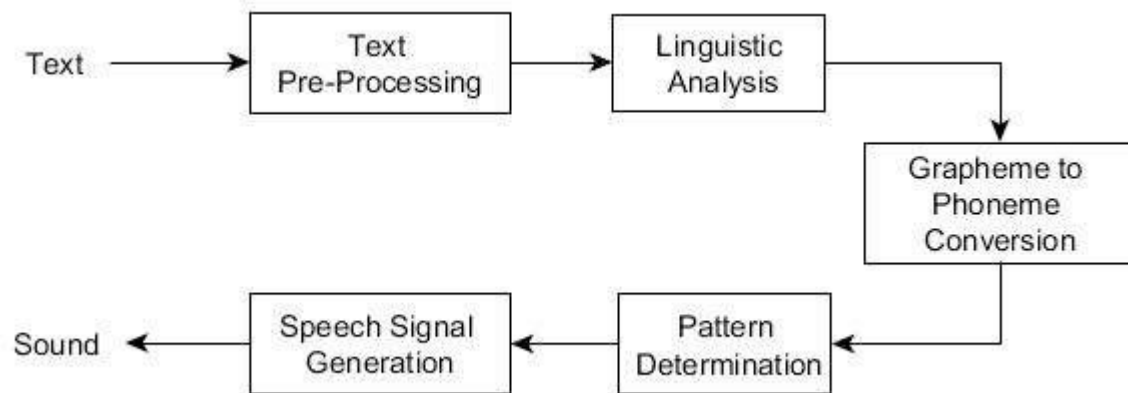
Whisper is an automatic speech recognition (ASR) system that has undergone extensive training by assimilating over 680,000 hours of multilingual, multitasking data sourced from the web. This comprehensive learning approach allows Whisper to exhibit proficiency in recognizing and transcribing speech across various languages and tasks. Findings suggest that use of this broad and diverse information can increase sensitivity to noise, background noise, and language nuances. In addition to improving security, the Whisper system has also demonstrated the ability to type in multiple languages and communicate from those languages to English. This open source project is intended to provide a basis for the development of practical applications and to support ongoing research in linguistics.



The Whisper architecture is a simple end-to-end approach based on the encoder-decoder Transformer. The audio input is split into 30-second segments, converted into a log-Mel spectrogram, and finally fed into the encoder. At the same time, the decoder works with approximate registers and provides unique symbols. These tokens organize a pattern to perform many tasks, including language recognition, multilingual typing, English translation.

Text to Speech - GTTS

Google Text-to-Speech (gTTS) is a powerful and easy-to-use Python library that converts text to speech. Developed by Google, the API leverages machine learning and advanced text-to-speech synthesis technology to deliver high-quality and expressive speech output. With gTTS, users can easily create audio files from text, making it easy to integrate audio capabilities into a variety of applications, including assistant programs, audio, navigation systems, and accessibility tools.



The simplicity of gTTS lies in its ease of use, converting text to speech with a few lines of code. Users can adjust parameters such as language, speaking rate, and volume to customize the output to specific preferences. Its accessibility and performance make gTTS a popular choice for developers looking for an easy-to-use and versatile text-to-speech solution that helps develop applications that require voice interaction.

Data consumption

The data collection process begins with recording audio feedback from end users, which is then converted into paper. To ensure the correctness of the data and reduce the risk of loss, this data is sent in continuous streams using Kafka messaging. Retrieval of data occurs dynamically and the processing of data takes place in the Kafka cluster.

The data obtained after consumption is fed to the trained language model (LLM) before using the general information in the user data set. Education LLM. This acts as a repository of content that supports the understanding of a particular site. Building on previous LLM implementations, it has been successful and has provided a positive impact on understanding and analysis by practical users. This end-to-end process enables comments to be translated into agreements, supporting efficient and informed decision-making.

EVALUATION

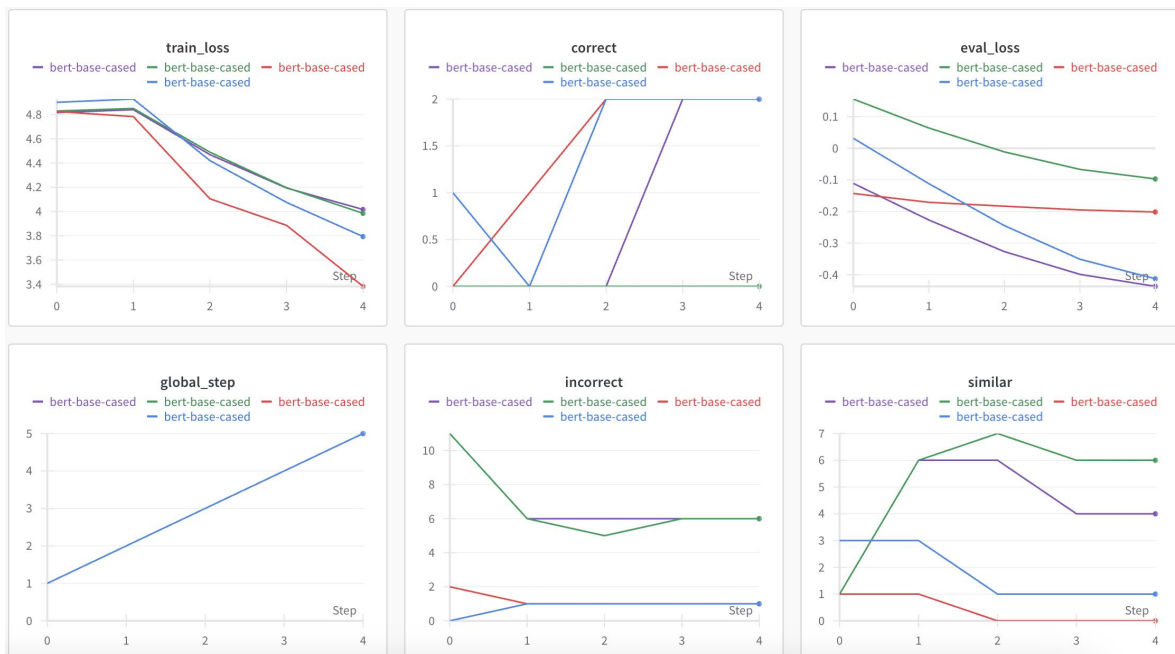
Question answer set (stored as a csv file)

	A	B
1	question	answer
2	Which Mohammedan criminal law were the local Governments guided by before 1827?	English criminal law
3	When was the Indian Penal Code introduced?	1860
4	What was the year when the judicial system of Bombay was revised?	1827
5	Which Presidencies were still following Mohammedan criminal law at the time of the Indian Penal Code's operation?	Bengal and Madras
6	In which year was the Indian Penal Code implemented in India?	1860

Dataset generated to train the model (stored as a json file)

```
[
  {
    "context": "which were partly removed by Regulations of the local Governments. In 1827 the judicial system of Bombay was thoroughly revised and 1 be investigated inquired into tried and otherwise",
    "qas": [
      {
        "id": "4",
        "is_impossible": true,
        "question": "Which Mohammedan criminal law were the local Governments guided by before 1827?",
        "answers": [
          {
            "text": "English criminal law",
            "answer_start": 127
          }
        ]
      }
    ]
  },
  {
    "context": "s 1 Title and extent of operation of the Code.This Act shall be called the Indian Penal Code and shall 3.extend to the whole of India local Governments. In 1827 the",
    "qas": [
      {
        "id": "5",
        "is_impossible": false,
        "question": "When was the Indian Penal Code introduced?",
        "answers": [
          {
            "text": "1860",
            "answer_start": 188
          }
        ]
      }
    ]
  }
]
```

Training Evaluation



OUTPUT

localhost:8501

Deploy

Voice Based Evaluation: Assessing Indian Law

Question: 1 What was the year when the judicial system of Bombay was revised?

[Click to Answer](#)

[Don't know answer](#)

Select answer duration (seconds):

10 100

Answer provided: 1827

Correct answer is: 1827

Question: 2 Which Presidencies were still following Mohammedan criminal law at the time of the Indian Penal Code's operation?

[Click to Answer](#)

[Don't know answer](#)

Select answer duration (seconds):

10 100

Answer provided: Madras and Bengal

Correct answer is: Bengal and Madras

localhost:8501

Finish update

Question: 9 Which Mohammedan criminal law were the local Governments guided by before 1827?

[Click to Answer](#)

[Don't know answer](#)

Select answer duration (seconds):

10 100

Answer provided: The English criminal law

Correct answer is: English criminal law

Question: 10 What is the aim of administering criminal law in Presidency towns before 1860?

[Click to Answer](#)

[Don't know answer](#)

Select answer duration (seconds):

10 100

Answer provided: Aim is to define offences and specifying punishments

Correct answer is: to define offences and specifying punishments

Total score is: 8 / 10

CONCLUSION

The Voice-based Evaluation: Accessing Indian Law system incorporating concepts of Natural Language Processing (NLP) and Large Language Models (LLMs) utilizing LangChain's Prompt templates and Mistral-7B-Instruct models for generating questions, and answer generation through Sentence Transformers, Vectorisation, embedding and vector stores with help of Google Flan models, and generating custom data sets and training those with BERT base models with help of SimpleTransformer techniques has demonstrated promising results. The use of wand technology has made the developer easy to coordinate and understand the training in a better way. The similarity checks using cosine similarity has given precise outcome with regard to answer evaluation.

The integration of speech-to-text technology, such as Whisper AI, has effectively facilitated the user interface and usability of user-provided answers in real-time. The system's performance aligns with expectations, showcasing the viability of voice-centric evaluations in assessing user responses. There is a clear potential for further enhancement to optimize user-friendliness and to refine the system into a more intricate interview-based model. The successful outcomes achieved thus far lay a solid foundation for continued advancements, encouraging the exploration of additional features and improvements that can contribute to the system's overall efficacy and user satisfaction.

DIRECTIONS FOR FUTURE WORK

There is great potential to extend and modify the current model to enable radio communications in a wide range of activities, including business and other activities. By leveraging the flexibility and adaptability of the model, we can improve the ability to engage with customers in a social and personal way. This model can be modified to intelligently generate recurring questions based on user experience and responses, rather than relying on predefined questions. This approach not only improves the user experience, but also tailors the interview process to better assess a person's depth of understanding and expertise in a field. The overall aim of the model is in line with the general goal of creating a more interactive and visual dialogue that meets the needs of professionals, making it diverse and more integrated and flexible.

BIBLIOGRAPHY / REFERENCES

- [1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, *"Attention Is All You Need"*, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.
- [2] Dongli Han, Takahiro Ohno, *"A Case Study on Experimental-data validation for Natural Language Processing"*, The 11th International Conference on Computer Science & Education (ICCSE 2016) Nagoya University, Japan
- [3] M. P. Bhuyan, S. K. Sarma, M. Rahman, *"Natural Language Processing based Stochastic Model for the Correctness of Assamese Sentences"*, IEEE Conference Record # 48766
- [4] Shojreh Rad Rahimi, Ali Toofanzadeh Mazhdehi, *"An Overview on Extractive Text Summarization"*, 2017 IEEE 4th International Conference on KBET
- [5] Michael Mohler, Rada Mihalcea, *"Text-to-text Semantic Similarity for Automatic Short Answer Grading"*, 12th Conference of the European Chapter of the ACL
- [6] Feng Zhang, Gaoyun An, Qiuqi Ruan, *"Transformer-based Natural Language Understanding and Generation"*, Proceedings of ICSP2022
- [7] Liliane do Nascimento Vale, Marcelo de Almeida Maia, *"Towards a question answering assistant for software development using a transformer-based language model"*, 2021 IEEE/ACM Thirs International Workshop on BotSE
- [8] ULDN Gunasinghe, WAM DE Silva, *"Sentence Similarity Measuring by Vector Space Model"*, International Conference on Advances in ICTer
- [9] Sazianti Mohd Saad, Siti Sakira Kamarudin, *"Comparative Analysis of Similarity Measures for Sentence Level Semantic Measurement of Text"*, IEEE International Conference on Control System
- [10] Yanni Li, Haisheng Li, Qiang Cai, Dongmei Han, *"A Novel Semantic Similarity Measure within Sentences"*, 2nd International Conference on Computer Science and Network Technology
- [11] Nityam Agarwal, Poorvi Seth and Merin Meleet, *"A New Sentence Similarity Computing Technique Using Order and Semantic Similarity"*, International Conference on Innovative Computing
- [12] Zhao Jingling, Zhang Huiyun, Cui Baojiang, *"Sentence Similarity Based on Semantic Vector Model"*, Ninth International Conference on P2P

COMPLETED CHECKLIST

S.No	Checklist	Status
1.	Is the Cover page in proper format?	Yes
2.	Is the Title page in proper format?	Yes
3.	Is the Certificate from the Supervisor in proper format? Is it signed?	Yes
4.	Is Abstract included in the Report? Is it properly written?	Yes
5.	Does the Table of Contents page include chapter page numbers?	Yes
6.	Does the Report contain a summary of the literature survey?	Yes
7.	Does the Report have Conclusion / Recommendations of the work?	Yes
8.	Are References/Bibliography given in the Report?	Yes
9.	Have the References been cited in the Report?	Yes
10.	Is the citation of References / Bibliography in proper format?	Yes