2024 6th International Symposium on Advanced Electrical and Communication Technologies (ISAECT) | 979-8-3315-2998-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISAECT64333.2024.10799776

On the Analysis of a BERT-based Domain-Specific Question Answering Models for Indian Legal System

Nisarg Shah
Department of Computer Science and
Engineering, School of Technology,
Pandit Deendayal Energy University
Gujarat, India, 382007
Email: nisarg.sce20@sot.pdpu.ac.in

Hiren Kumar Thakkar
Department of Computer Science and
Engineering, School of Technology,
Pandit Deendayal Energy University
Gujarat, India, 382007
Email: hiren.pdeu@gmail.com

Hiren Mewada
Electrical Engineering Department,
Prince Mohammad Bin Fahd University,
P.O. Box 1664, Al Khobar 31952,
Kingdom of Saudi Arabia
Email: hmewada@pmu.edu.sa

***Abstract*—The Indian legal system is too large and complex which includes the largest Constitution in the world, thousands of laws, central laws, thousands of acts and many bilateral, regional, and multilateral treaties and agreements with different countries. This makes it difficult to understand and memorize the contents for legal professionals as well as the general public. Some individuals and government officials take advantage of this fact by doing unlawful acts and avoid punishments due to the unawareness in public. So, we are developing a closed domain extractive model for question answering on the Indian legal system. The motivation behind developing a Question Answering System (QAS) on Indian laws is to enhance the accessibility of legal information for both legal professionals and the general public. PDFs of legal acts were collected from the Government website from which Question Answering dataset was created. Dataset contains _ automatic system generated questions using T5 question generated model and then manual cleaning such as removing wrong questions, minor changes were done. Models used for QA are DistilBERT, RoBERTa, BERT Large. BERT Large outperforms the other models with cosine similarity of 0.944, BLEU score of 0.781, Exact match(EM) with 0.72. Question answering model provides short and precise answers of the questions with context from which the answer was extracted to avoid giving wrong information.**

***Index Terms*—Indian legal system, Question Answering system(QAS), BERT, T5**

I. Introduction

The Indian legal system includes the Constitution, thousands of laws and acts, many bilateral, regional, multilateral treaties and agreements with different countries [1]. It can be difficult for the general public, academics and legal professionals to navigate through this. This arises a need for effective tools that can provide quick and accurate access to legal knowledge.

This need is satisfied by a question-answering (QA) system for Indian laws which gives users clear and concise answers to their questions. These systems use machine learning and natural language processing (NLP) to extract accurate responses from large legal texts based on the context [2].

Question Answering (QA) system in NLP is used for information retrieval in which it automatically answers to the questions asked by humans using either a pre-structured database or a collection of natural language documents [3]. Fig. 1 shows the evolution of the question-answer system.

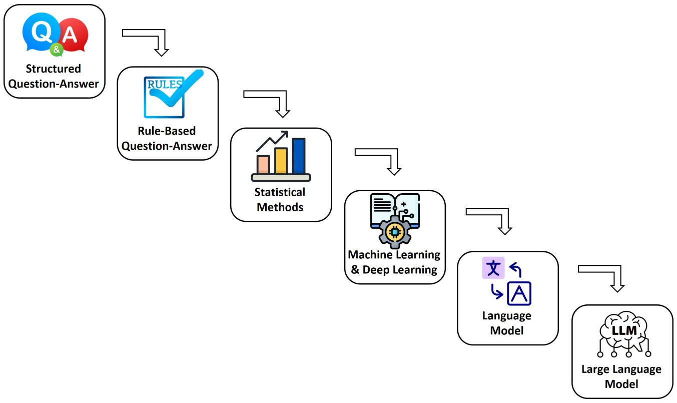

Q
A
Structured Question-Answer
RULES
Rule-Based Question-Answer
Statistical Methods
Machine Learning & Deep Learning
Language Model
LLM
Large Language Model

Fig. 1: The Evolution of Automatic Question-Answer System

A. History of Question-Answer System

- **Structured Question Answering(QA) [4]**: Database store in a structured means table form. To answer the questions, a query was generated in SQL or another language. Questions were limited, answers were most likely single word or number and it is very difficult to create a query for complex questions. Bank, medical staff and many store data in Structured way since it is easy to find the answers from it.
- **Rule-based Question Answering(QA) [5]**: Rule-based Question Answering is a system where an answer is generated or extracted based on predefined rules. Rules are generally created to find specific patterns from the text such as questions regarding the amount of money can be answered by finding the $ or Rs. sign indicating the money.
- **Statistical Question Answering(QA) [6]**: It uses statistical methods(Using probabilities) to understand and answer the question. It is generally used for information retrieval such as finding relevant documents or classifying

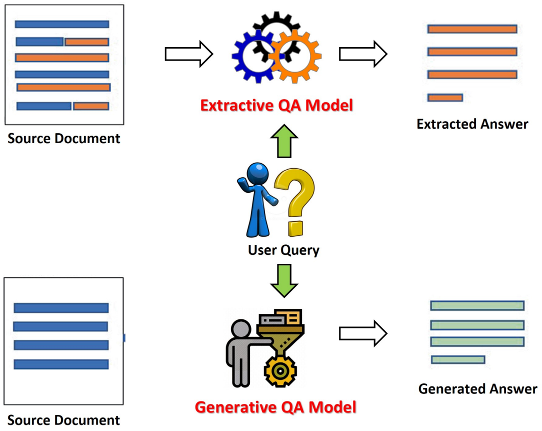


Fig. 2: Extractive Vs Generative QA System.

documents in one of the categories, predicting or generating next word.

- **Machine Learning(ML) and Deep Learning(DL) [7]**: Here, unlike previous methods computers learn from the data and try to predict the answer to the question. They use advanced algorithms and neural networks to understand questions and generate the answer. Regression means predicting future values, image as well as text classification is possible but they have limitations regarding understanding and interpreting context.
- **Language model**: In 2017, transformers were developed which were able to process sequential data such as text which allows to capture complex relations within data leading to efficiently answer the questions. GPT-1, BERT are examples of pre-trainend language models. They are trained on large amounts of unlabelled data enabling them to understand patterns and structure of the text.
- **Large language model**: They have gained popularity due to their understanding of human-like text, image, video, audio. ChatGPT is a generative well-known Large language model(LLM).

B. *Types of Question-Answer System*

Based on the method to find the answers to the questions we can divide the QA system into 2 types Extractive Question-Answering (QA) and Generative Question-Answering (QA). The Fig. 2 shows the difference between the Extractive QA and Generative QA.

1) **Extractive QA**: It involves finding a specific portion of text within a given context such that it can answer the asked question. It retrieves exact phrases or sentences. Models such as BERT (Bidirectional Encoder Representations from Transformers), RoBERTa (A Robustly Optimized BERT Pretraining Approach), ALBERT (A Lite BERT), DistilBERT are used for Extractive answering.
2) **Generative QA**: It creates answers by generating text, rather than extracting it from a predefined source. This involves creating a new response based on logic and reasoning. Models such as OpenAI's GPT-3.5 and GPT-4 are used for Generative answering.

Based on the scope of the domain from which the model can answer the question the QA system can be divided into 2 types Open domain QA and Closed domain QA.

1) **Open-domain QA systems**: They can answer questions from a large variety of topics without being restricted to a specific domain or subject area. ChatGPT, virtual assistants such as Siri, Google Assistant, or Alexa are systems based on open-domain.
2) **Closed-domain QA systems**: They are designed to answer questions within a specific domain or area. They are more specialized on a specific domain. Chatbots used in medical, e-commerce websites, the banks are based on closed-domain systems. We implemented Extractive closed domain (Indian legal system) QA.

This paper investigates the creation and application of a QA system that is especially suited for Indian legal acts. We will look into dataset generation,fine tuning, and answer finding from PDF collection.

C. *Example of Question-Answer System*

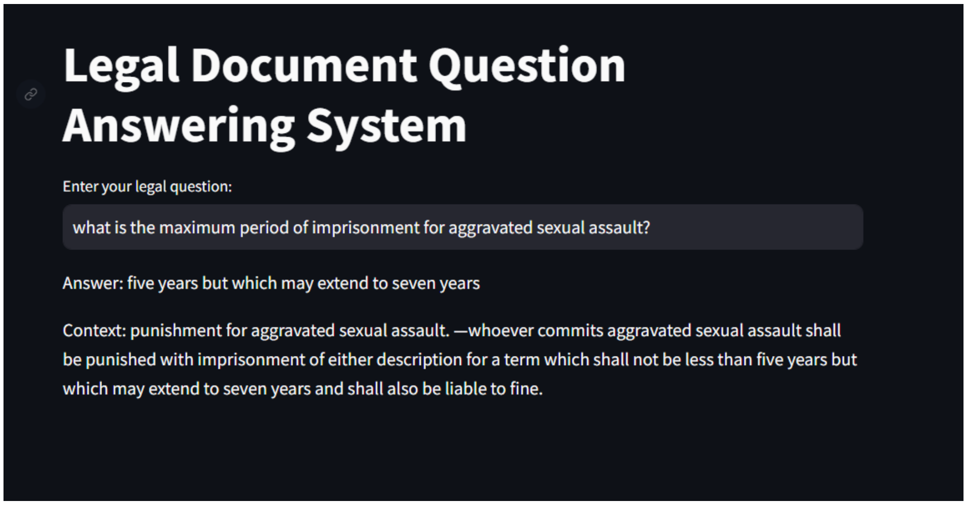
Fig. 3 shows an example of a question answering system developed for the Indian Legal system. Question is provided and using it a short, precise answer is given with the context so that the user can verify the answer and giving wrong information can be avoided.

II. Related Work

Natural language processing (NLP) experts have been paying close attention to recent developments in Legal Question Answering (LQA) systems, especially with regard to the use of deep learning models to improve system performance. Due to their intricate linkages and specialised vocabulary, legal documents pose certain issues that call for specialised answers. An extractive methodology is commonly used by traditional QA systems, in which responses are taken straight from the settings that are supplied. But the attention has switched to generative models, which use large corpora to generate replies that are more sophisticated, since the introduction of Large Language Models (LLMs) such as GPT-3 and Legal-BERT.

The study conducted by Nigam et al. (2023) [8] highlights the challenges associated with modifying AI models for the Indian legal domain, which is characterised by its distinct legal language and structure. The researchers investigate different embedding models and quality assurance systems, assessing their effectiveness through the application of syntactic and semantic metrics in addition to expert legal evaluations. The study also highlights the potential of language modellers (LLMs) such as GPT-3 in producing precise answers that are customised to the Indian legal context, underscoring the significance of embedding models in generating answer.

Prabhu et al.'s (2024) [9] work explores unsupervised approaches to legal question answering with an emphasis on problems related to U.S. Civil Procedure. To handle the intricacy of legal documents, their method combines transformer models, such as Legal-BERT, with word embeddings. Notably, their work improves the system's capacity to handle



Legal Document Question Answering System

Enter your legal question:

what is the maximum period of imprisonment for aggravated sexual assault?

Answer: five years but which may extend to seven years

Context: punishment for aggravated sexual assault. —whoever commits aggravated sexual assault shall be punished with imprisonment of either description for a term which shall not be less than five years but which may extend to seven years and shall also be liable to fine.

Fig. 3: A Typical Example of a Question-Answer System

long legal explanations by introducing a unique multi-level summarisation approach employing T5 transformers. Increasing performance measures such macro F1 scores has been demonstrated to be possible with this strategy, especially when used on difficult legal datasets.

To sum up, the increasing trend of using LLMs and deep learning approaches for legal Question Answering emphasis on tailoring these models to the particular needs of the legal domain. The advancement of legal QA systems is contingent upon the incorporation of summarisation approaches and the investigation of diverse embedding models. Together, these efforts advance our knowledge of the difficulties and bring possibilities, opening the door for future advancements in more efficient and contextually appropriate systems. However,there are lots of risks also involved in using advance approaches. The first and foremost is to misunderstand the meaning behind it. Lawyers frequently interpret legal context in various ways during case arguments, therefore it is crucial to focus on every word before presenting answer. In addition, due to complex language, there are higher chances to misinterpret even for humans so, we are planning for extractive type only.

III. MODELS

There are multiple models when it comes to question answering such as GPT,T5, BART, Pegasus for generative QA and BERT and its variants, XLNet for extractive QA. The reasons for using BERT and its versions are explained in ensuing paragraphs.

1. First of all, BERT follows Transformer based architecture which relies on self-attention mechanism. It is a technique that allows a model to focus on different parts of input and thus enabling to capture relationships between words over long distances.
2. Additionally, BERT is a Bidirectional model which means it is able to reads text from both left and right simultaneously, enabling to understand a word's whole context in a phrase.
3. In addition, BERT is trained on a large dataset of 800M words from BooksCorpus and 2,500M words from English Wikipedia using Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) [10]. This gives BERT a high understanding of different domains and can be fine tuned for a specific domains with high accuracy.

We have fine tuned 3 BERT variant language models DistilBERT, RoBERTa and BERT Large. Here we will provide information regarding these models such as training parameters, dataset they were trained on, as well as their evaluation parameters from the training.

1) **BERT Large** : Google AI Language created a configuration of BERT (Bidirectional Encoder Representations from Transformers) called BERT Large. It has 24 layers which are composed of feedforward neural networks and self-attention processes. With a hidden size of 1024 and about 340 million parameters, BERT Large. It got exact Match(EM) on squad_v2 validation set as 80.885 and F1 score as 83.876.
2) **DistilBERT** [11] : It is a more compact and effective version of the BERT architecture. It is 60% quicker and uses 40% less parameters than bert base model also nearly maintaining BERT's performance by 97%. Unlike BERT large, which has 24 layers, DistilBERT has six levels. The model achieves an F1 score of 86.9 on the development set of dataset SQuAD v1.1.
3) **RoBERTa** : "A Robustly Optimised BERT Pre training Approach," or "RoBERTa," is a sophisticated version of the BERT (Bidirectional Encoder Representations from Transformers) paradigm. It was developed by Facebook AI Research (FAIR). Next sentence prediction (NSP) and masked language modeling (MLM) are the two tasks in BERT's pretraining but RoBERTa eliminates NSP, focusing solely on MLM. it's hyperparameters during training are as shown in Table I. It achieved an exact

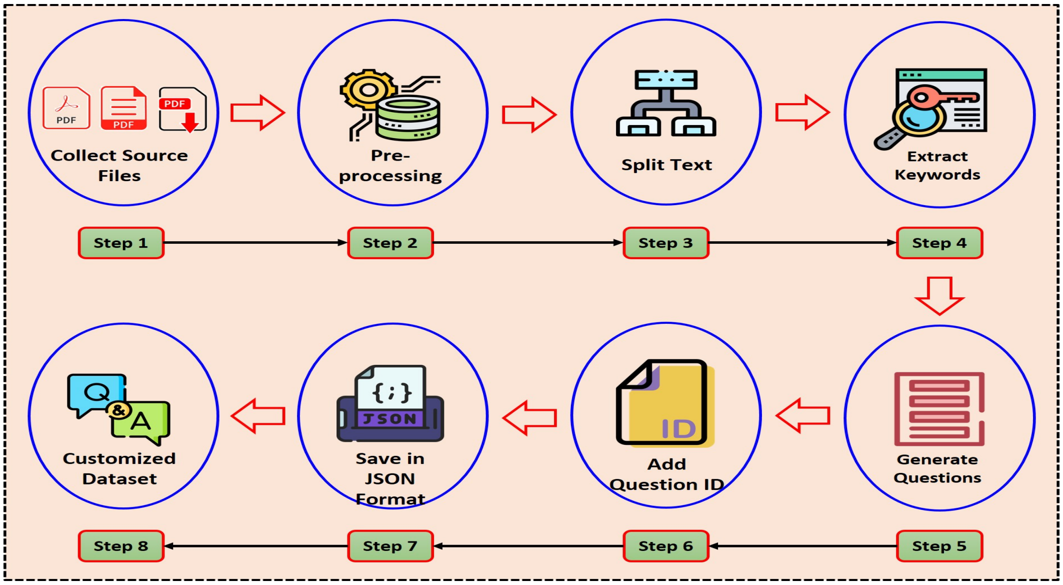


Fig. 4: The Process of Customized Dataset Generation for Question-Answer System.

score of 79.87 and f1 score of 82.91.

TABLE I: List of Hyperparameters used during the model training.

Hyperparameter	**Value**
batch size	96
max seq len	386
learning rate	3e-5
number of epochs	2
warmup proportion	0.2
lr schedule	Linear Warmup
max query length	64
doc stride	128

IV. Methodology

A. Dataset Generation

The process of customized dataset generation for question-answer system is shown in Fig. 4.

1) First of all, We need to collect required PDFs on laws and acts to generate the dataset. We have collected different pdfs on acts from the official Govt. website. The list of prominent Indian Laws are described in Table II. After that we converted it into string form for better accessibility. These are the PDFs which are used to generate dataset using reference from [8].
2) Pre-processing such as removing title (It is extracted using 'act' keyword since all PDFs are acts), index(find 2nd appearance of title and remove in between all which is index), footer, removing punctuation other than '.','-' since '.' is used to get sentences and '-' is used in many text for mentioning dates.
3) Then text is split into chunks of 300 or more characters which we will use for context in the dataset.

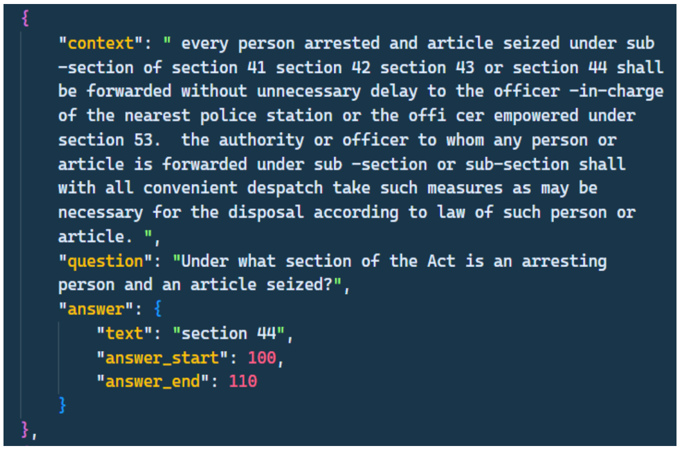
TABLE II: List of Acts in Indian Law.

S.No.	Act
1	Criminal Procedural Code
2	Arms Act
3	Extradition Act of 1962
4	Contempt of Courts Act
5	Protection of Children from Sexual Offences Act
6	Indian Evidence Act
7	Unlawful Activities Prevention Act
8	Criminal Procedure Identification Act
9	Narcotic Drugs and Psychotropic Substances Act
10	Prevention of Money Laundering Act
11	Information Technology Act
12	Prevention of Corruption Act of 1988
13	Gram Nyayalayas Act of 2008
14	Indian Penal Code
15	Prisons Act of 1894

4) Answers are extracted using named entity recognition, words with entities of either of LAW, DATE, ORG are extracted. Also, store indexes of answers in context which is required when fine tuning the model.
5) Give these words and sentences to Google's T5 question generation model. It will generate questions based on sentence and answer.
6) We then manually removed some questions added due to irregular spaces and inaccuracy of the model.
7) Then, provide Unique id to every pair of context, question, answer and store it in JSON file format
8) save this file for later fine tuning.

The description of the generated dataset is as follows:

- **File format**: JSON which stores information in key, value pair format. Every object stores context, question, answer which are explained below in detail.
- **context**: It is a 300 or more characters long text which contains the answer for the provided question.
- **Question**; It is the question generated by the Google's



```
{
    "context": " every person arrested and article seized under sub
    -section of section 41 section 42 section 43 or section 44 shall
    be forwarded without unnecessary delay to the officer -in-charge
    of the nearest police station or the offi cer empowered under
    section 53.  the authority or officer to whom any person or
    article is forwarded under sub -section or sub-section shall
    with all convenient despatch take such measures as may be
    necessary for the disposal according to law of such person or
    article. ",
    "question": "Under what section of the Act is an arresting
    person and an article seized?",
    "answer": {
        "text": "section 44",
        "answer_start": 100,
        "answer_end": 110
    }
},
```

Fig. 5: A Snapshot of a Customized Dataset Generated for Legal Question-Answer System.

T5 question generation model.

- **Answer**: It contains 3 parameters answer and its starting, ending indexes. Model actually only uses indexes however we have included text of the answer for human references.
- **Total of context-question-answer pairs**: 2087

The Fig. 5 shows an example of the dataset format which contains question, answer with its start and end indexes, context from which answer is extracted, id to distinguish the pairs.

B. Fine tuning the model

The Process for fine tuning is referenced from [12] which is explained as follows.

1) Load dataset and split it into train, test, validation by 70,15,15 ratio.
2) Import model and tokenizer from Hugging face models and take care of the following points:
   - Specify maximum length as per the model such as 512 for BERT large model.
   - Specify truncation = ”only_second” to avoid scenarios where examples may have a very long context that exceeds the maximum input length of the model.
   - Next, map the start and end positions of the answer by setting return_offset_mapping = True.
   - Now it is possible to find the start and end tokens of the answer to the question and which corresponds to the context Using sequence_ids method.
3) Then, use DefaultDataCollator to create a batch of examples which can be imported from transformers.
4) Then, specify training arguments required to fine tune the model. Then, pass model, arguments, dataset, tokenizer, and data collator to Trainer.
5) I have use batch_size = 16, learning_rate = 2e-5, epochs = 3, weight_decay = 0.01.
6) Evaluate the model on the test dataset and save the model for future references.

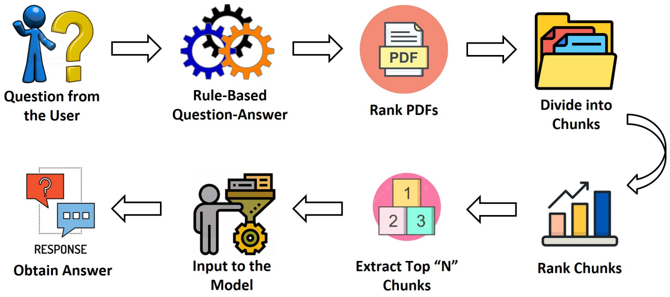


Fig. 6: The Detailed Process of Generating Answer for a Given Prompt.

C. Question-based Answer Genaration

The detailed process of generating answer from a given prompt is shown in Fig. 6.

It is the combination of document retrieval, text retrieval and answer extraction. The whole process of finding an answer from available PDFs is mentioned here [13]:

1) questions related to the Indian legal system will be provided by the user. It should be answerable from the database of PDFs.
2) tokenized the question as well as pre-process the PDF text before ranking them. pre-process steps includes removinf index, title, footer from the text
3) Now, Rank all PDFs using BM25+ algorithm (other algorithms can also be used) and get top 3 ranked documents to reduce time and avoid wasting of needless computation.
4) Get text from the top 3 documents using TF-IDF algorithm.
5) Now, split text into chunks of 200 or more characters with 50 overlapping characters. So that answer getting separated into another chunk can be avoided.
6) Rank all chunks using the same BM25+ algorithm and get the top 5 chunks. So that question answering model have less context to find the answer from. This can improve the accuracy of the model.
7) Provide questions and context to model for answers. There is a chance that answer got missed during ranking PDFs or chunks as well as the model finding the wrong answer. To avoid users getting wrong information we also provide context from which the answer was extracted so that the user can verify the answer in case of doubt.

V. Results and Discussion

Results of training, validation and testing are mentioned below.

Due to limited computing power, we only ran models for 3 epochs. Fig. 7 shows the performance evaluation of a models for the training data. As shown in Fig. 7 RoBERTa had the least training loss of only 0.753, followed by DistilBERT and BERT-Large of 0.99 and 1.1, respectively. For the validation, every model had almost same loss with RoBERTa having the lowest of 1.2 and BERT-large having the highest of 1.29 as shown in Fig. 8.

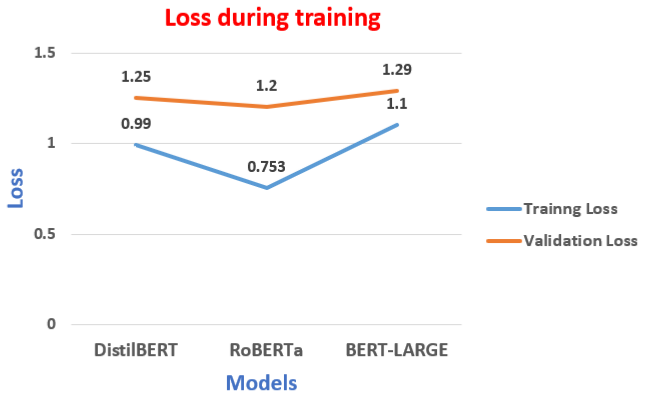


Fig. 7: The Performance Evaluation of a Proposed Model for the Training data.

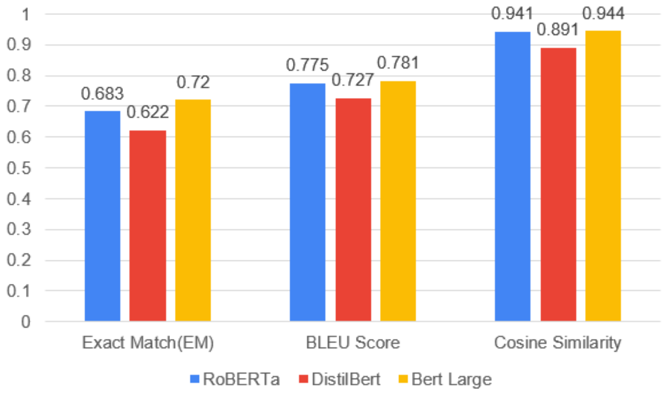


Fig. 8: The Performance Evaluation of a Proposed Model for the Test data.

1) **Exact Match (EM)**: Exact match is a measure used to determine if prediction and reference answers are exactly the same or identical. The equation to calculate the EM is shown in Eq. 1. DistilBERT has 0.622, RoBERTa has 0.683 and BERT Large has 0.72 EM score.

$$\text{Exact Match} = \frac{count(Y\_pred = Y\_actual)}{Total\_no.\_of\_pedictions} \quad (1)$$

Here, Y_pred and Y_actual is the prediction made by the proposed model and the ground truth, respectively.

2) **BLEU Score**: The BLEU score helps to know the quality of a machine translated text. The BLUE score is calculated by comparing the number of n-grams in the predicted text that matches with the reference translations as shown in Eq. 2. Here, only one word is compared since nearly 20% of answers are one word. DistilBERT has 0.727, RoBERTa has 0.775 and BERT Large has 0.781 EM score.

$$BLEU\ Score = BP \times exp(\sum_{n=1}^{N} W_n log P_n) \quad (2)$$

Here, the Brevity Penalty (BP) is a parameter which penalizes predicted answers shorter than the actual answer. It can be calculated as shown in Eq. 3. P_n is the geometric average of the modified n-gram precision and w_n is the weight, where $w_n > 1$.

$$BP = \begin{cases} 1, & \text{if } c > r; \\ \mathrm{e}^{(1-r/c)}, & \text{if } c \leq r \end{cases} \quad (3)$$

Where:

- r is the length of the reference translation (or closest reference in length if there are multiple).
- c is the length of the candidate translation.
- P_1 is the precision as shown in Eq. 4.

$$P_1 = \frac{Number\_of\_matching\_unigrams}{Total\_number\_of\_words\_in\_candidate} \quad (4)$$

- The BP is applied as shown in Eq. 3, which penalizes shorter candidate translations.

3) **Cosine Similarity**: Text is transformed into vectors and then cosine similarity is used to measure similarities between them as shown in Eq. 5. It is independent of the length of the answers and reference. This gives a better comparison between them. DistilBERT has 0.891, RoBERTa has 0.941 and BERT Large has 0.944 EM score.

$$\text{Cosine Similarity} = \frac{A.B}{||A|| \times ||B||} \quad (5)$$

Where:

- A = vectorized form of prediction,
- B = vectorized form of actual value

The BERT Large model clearly gives better test results despite having the most training and validation loss compared to other models with 0.72 Exact Match (EM), 0.781 BLEU Score and 0.944 Cosine similarity which is clearly visible in Fig. 7 and Fig. 8. From the results, it is implied that the pre-trained models such as RoBERTa, DistilBERT, and BERT-Large can be considered as a starting point to address the problem of question-answer in a specific domain such as legal after fine-tuning on the domain specific dataset. However, the performance of such models can be further improvised after further investigations.

VI. Conclusion and Future Scope

In this paper, We analyzed the process to design the question answering system including how to train pre-trained models. The study also focuses on the aspects of different text preprocessing techniques such as removing stop words, lemmatization, removing index, footer, header, algorithms such as BM25+, TF-IDF, Dataset generation using T5 question generation model, evaluation of models using Exact Match, cosine similarity, BLEU Score, and generating the answer from PDFs. Furthermore, Question answering system provides enhanced accessibility means instead of going through countless documents, we just need to provide the query and a short and precise answer will be given. This will be helpful for both general public and legal professionals as both have to know the law which governs India and it affects both. In addition, We ensure the correctness of the answer by providing context from which the model extracts the answer to cross check it. Also, a user-friendly interface makes it easy for users to have the best experience. However, there are many problems associated with this QA system such as laws are constantly changing, new ones are coming while old ones get improved or removed, if the system is connected through Govt. official website of laws then system can automatically remove or add documents making it completely automatic. Another problem is due to its extractive nature resulting in inability of answering Yes/No or outside of context answers which are very common. We can maintain a dictionary of synonyms to easily find the best answer from the context. This question answering model development method can also be applied into other fields such as medical, customer support, education, tourism and many others.

References

[1] M. Blackhawk, "Federal indian law as paradigm within public law," *Harvard Law Review*, vol. 132, no. 7, pp. 1787–1877, 2019.

[2] A. Zafar, S. K. Sahoo, H. Bhardawaj, A. Das, and A. Ekbal, "Ki-mag: A knowledge-infused abstractive question answering system in medical domain," *Neurocomputing*, vol. 571, p. 127141, 2024.

[3] R. Goyal, P. Kumar, and V. Singh, "Automated question and answer generation from texts using text-to-text transformers," *Arabian Journal for Science and Engineering*, vol. 49, no. 3, pp. 3027–3041, 2024.

[4] A. Frank, H.-U. Krieger, F. Xu, H. Uszkoreit, B. Crysmann, B. Jörg, and U. Schäfer, "Question answering from structured knowledge sources," *Journal of Applied Logic*, vol. 5, no. 1, pp. 20–48, 2007.

[5] R. H. Gusmita, Y. Durachman, S. Harun, A. F. Firmansyah, H. T. Sukmana, and A. Suhaimi, "A rule-based question answering system on relevant documents of indonesian quran translation," in *2014 International Conference on Cyber and IT Service Management (CITSM)*, pp. 104–107, IEEE, 2014.

[6] M. A. C. Soares and F. S. Parreiras, "A literature review on question answering techniques, paradigms and systems," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 6, pp. 635–646, 2020.

[7] T. Tahsin Mayeesha, A. Md Sarwar, and R. M. Rahman, "Deep learning based question answering system in bengali," *Journal of Information and Telecommunication*, vol. 5, no. 2, pp. 145–178, 2021.

[8] S. K. Nigam, S. K. Mishra, A. K. Mishra, N. Shallum, and A. Bhattacharya, "Legal question-answering in the indian context: Efficacy, challenges, and potential of modern ai models," 2023.

[9] M. Prabhu, H. Srinivasa, and A. Kumar, "SCaLAR NITK at SemEval-2024 task 5: Towards unsupervised question answering system with multi-level summarization for legal text," in *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)* (A. K. Ojha, A. S. Doğruöz, H. Tayyar Madabushi, G. Da San Martino, S. Rosenthal, and A. Rosá, eds.), (Mexico City, Mexico), pp. 193–199, Association for Computational Linguistics, June 2024.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[11] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," in *NeurIPS EMC*^{2}$Workshop$, 2019.

[12] H. face, "Question answering." https://huggingface.co/docs/transformers/tasks/question_answering, 2024. Accessed: 10-May-2024.

[13] J. Alzubi, R. Jain, A. Singh, P. Parwekar, and M. Gupta, "Cobert: Covid-19 question answering system using bert," *Arabian Journal for Science and Engineering*, 06 2021.