LONTAR KOMPUTER VOL. 15, NO. 3 DECEMBER 2024 **p-ISSN 2088-1541**
DOI : 10.24843/LKJITI.2024.v15.i03.p06 **e-ISSN 2541-5832**
Accredited Sinta 2 by RISTEKDIKTI Decree No. 158/E/KPT/2021

QnA Chatbot with Mistral 7B and RAG method: Traffic Law Case Study

Muhammad Roiful Anam[a1], Agus Subhan Akbar[a2], Heru Saputro[b3]

[a]Information Systems, University Islam Nahdlatul Ulama Jepara
Jl. Taman Siswa Pekeng, Tahunan, Jepara, Jawa Tengah, Indonesia
[1]muhammadroifulanam2001@gmail.com
[2]agussa@unisnu.ac.id (Corresponding author)
[3]herusaputro@unisnu.ac.id

Abstract

Mistral 7B is a language model designed to achieve high efficiency and performance in handling Natural Language Processing (NLP). This research will evaluate the model's effectiveness in legal data processing using the Retrieval-Augmented Generation (RAG) method, focusing on road traffic and transportation law No 22/2009. The system was built using the LangChain framework, followed by fine-tuning the model and evaluated using BERTScore. Results showed that the fine-tuned Mistral 7B achieved an F1 score of 0.9151, higher than the version without fine-tuning (0.8804) and GPT-4 (0.8364). To improve accuracy, the model utilizes specific keywords that make it easier to find relevant data. Fine-tuning was shown to enhance precision, while the use of key elements in questions helped the model focus more on important information. The results are expected to support the development of artificial intelligence (AI) in Indonesia's legal system and provide practical guidance for applying AI technology in other areas of law.

Keyword: *Mistral 7B, Retrieval-Augmented Generation, Fine-Tuning, AI, BERTScore*

1. Introduction

In the increasingly advanced digital era, AI technology has become essential in various fields[1]. One of these fields is law. With the complexity of legal data processing and the increasing volume of data, such as laws, regulations, and jurisprudence, there is an urgent need to create a system that can access, analyze, and present legal information efficiently and accurately. However, the answers produced by this AI system must still be accounted for [2], especially in the context of legal and ethical responsibility. This is important considering the potential consequences of AI decisions in processing and conveying information, which can have fatal consequences if misinterpreted.

Mistral 7B was introduced as a language model designed with high efficiency and performance to meet the challenges in Natural Language Processing (NLP). It offers competitive performance while maintaining high computational efficiency[3]. The advantages of this approach are in line with the innovations introduced by Transformer-based models, such as BERT and GPT, which significantly improve the ability to understand context and relationships between words[4]. One of the main advantages of Mistral 7B is its ability to handle large-scale text, both for analysis and relevant text generation. Mistral 7B is integrated with the Retrieval-Augmented Generation (RAG) method, combining retrieval and generation approaches to maximize effectiveness. This approach allows the system to compare the generated output with one or more references before generating an accurate answer[5].

This study aims to evaluate the effectiveness of the Mistral model in processing legal data, especially in Law Number 22 of 2009 concerning Road Traffic and Transportation. Law Number 22 of 2009 was chosen as the primary dataset because of its relevance to traffic and transportation regulations in Indonesia and its complexity, which requires in-depth analysis. By combining the Mistral model with the RAG method, this study is expected to significantly contribute to the development of AI technology in tracing, analyzing, and understanding legal documents more efficiently.

In addition, this study compares the Mistral 7B model, the fine-tuned Mistral, and GPT-4.o to identify which model is most effective in processing legal data. This study seeks to answer the main challenges in legal data processing, namely ensuring that the information presented by AI is accurate, contextual, and relevant. The results of this study are expected to be the basis for further development in the integration of AI technology in the Indonesian legal system and provide practical guidance for applying AI in other legal contexts.

2. Research Methods

The research begins by preparing and configuring the required software with the stages shown in Figure 1. The stages include installing the LangChain framework and several supporting libraries, such as Transformers for Mistral 7B model access, Accelerate to speed up training and inference using the GPU, Bitsandbytes to reduce memory usage, and Gradio to build the user interface. LangChain is the primary framework that integrates the large language model (Mistral 7B) with other components, such as the retriever in the RAG system. After the installation, the Mistral 7B model and tokenizer are initialized to process the text. The tokenizer breaks down the text input into tokens that can be processed by Mistral 7B as the core of the system responsible for understanding the question, analyzing the context, and generating answers. The dataset is then processed using the Retrieval-Augmented Generation (RAG) method, where the text is broken into chunks and converted into a numerical representation (embedding) to facilitate semantic search by the retriever.

The resulting embedding is stored in the Chroma Database, which serves as a retriever to match relevant data based on user queries. A text pipeline was built to integrate the context retrieved by the database with the generative capabilities of Mistral 7B. LangChain also organizes the system's workflow using prompt templates, ensuring the generated answers are formatted and contextualized to suit the user's needs. The primary function of Mistral 7B is to create context-based answers derived from the retriever. At the same time, RAG ensures relevance by retrieving appropriate chunks of text from the dataset. This involves combining two main capabilities: retrieving relevant data and generating answers for the final response. The process is accessed through the Gradio interface, which allows users to input questions and receive answers.

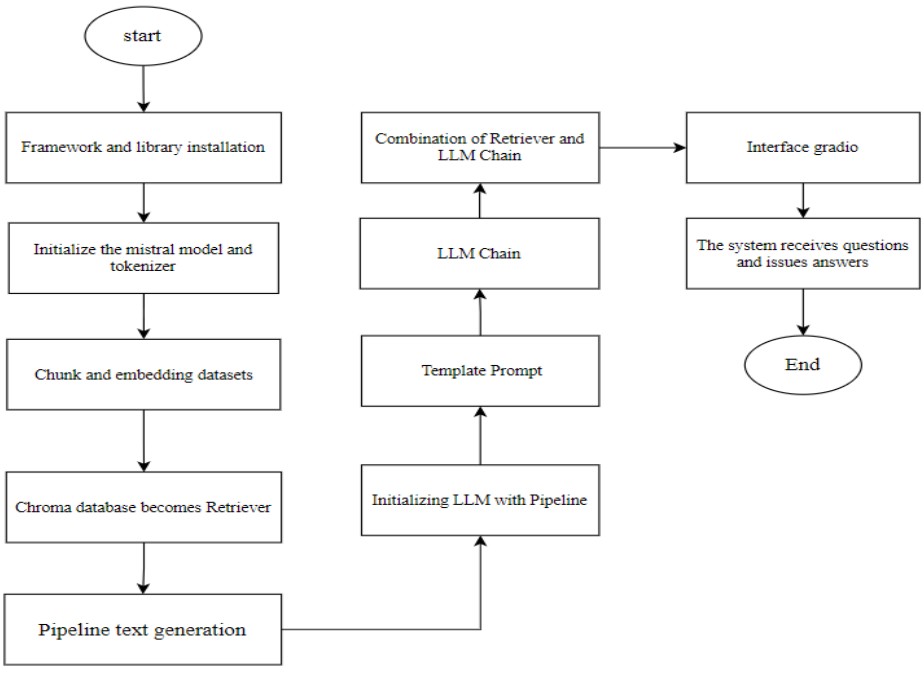

Figure 1. System Metodology

2.1. Dataset

This research dataset was obtained from the official website of the Korlantas Polri (https://korlantas.polri.go.id) in the form of a PDF file of Law Number 22 of 2009 concerning Road Traffic and Transportation, which was then converted into CSV format for further processing. The data in CSV format consists of two columns, namely keywords or questions and answers in the form of the contents of the law. The data is taken randomly from each chapter representative. The selection of these documents is based on the complexity of the rules that pose a challenge in evaluating the model's ability to answer legal questions. The question-answer pairs cover essential aspects, such as penalties, regulations, vehicle classifications, and specific conditions.

2.2. Language Model

The language model used in this study is Mistral-7B-Instruct-v0.3. This model is the third version of the Mistral Instruct series, significantly improving language comprehension and generation capabilities over previous versions. One of the main advantages of this model is the use of a more complete vocabulary and support for Tokenizer v3, which contributes to improved performance in understanding and generating text.

2.3. Retrieval-Augmented Generation (RAG)

The Retrieval-Augmented Generation (RAG) method is applied in this study to improve the accuracy and relevance of answers generated by language models. Retrieval-augmented generation (RAG) is a technique in natural language processing that combines generative models and retrieval models to improve the quality and relevance of the generated text [6]. In the RAG method, the retrieval model first searches for specific information from the knowledge base using semantic search, and then the generative model uses that information to generate more relevant and contextualized answers. This approach can reduce errors such as “hallucination” in AI models and help generate more appropriate text for tasks such as summarizing and answering questions in the context of this research. When a user asks a question, the RAG system searches and identifies the parts of speech most relevant to the given keywords.

After the search stage finds relevant information, the following process is text generation, which involves using language models to generate appropriate text[7]. At this stage, the language model is used to read and analyze the information found and compose answers that match the questions asked. The language model not only retrieves information directly from the document but also integrates existing knowledge in its memory to produce a more comprehensive and contextualized answer. Thus, this stage of answer generation ensures that the response provided is factual and adapted to the context of the question asked.

Two essential concepts support this process when applying RAG: vector and embedding. In the context of natural language processing, Vectors are numerical representations that transform text into a format that machine learning models can process to understand its meaning[8]. This representation allows the system to process text in a form that is easier to understand. Using vectors, the system can measure the similarity or relevance between a given keyword and the information in the data set. This helps determine which chunks of text are most relevant to return to the language model.

Embedding is a technique that converts text such as words, sentences, or documents into numerical vectors to help machine learning models understand the context and meaning of the text [9]. Embedding is designed so that words with similar meanings have similar vectors. For example, the words “I” and “me” will have almost the same vector because they have identical meanings. This embedding technique allows the model to understand the relationship between words in a broader context, thus providing more accurate and relevant answers. Embedding also helps minimize interpretation errors if the model does not recognize similar words or phrases as synonyms.

For example, when a user asks a question, the system will identify the most relevant parts of the document through semantic vector matching. Once the required information is found, the generative model will create a more contextualized and accurate answer. Figure 2 shows how the system processes legal documents to generate relevant answers. This process starts with the Document Separation Process, where legal documents are separated into small parts, such as

paragraphs. Once the document is separated, chunking techniques divide the text into smaller units, such as sentences or phrases, making it easier to process and convert to numerical representations. The chunks of text are then converted into numerical vectors using the Embedding Model, which creates vectors that reflect the semantic meaning of the text. These vectors are stored in the Vector Database, enabling semantic search to match user queries with relevant text sections.

When a user asks a question, the system translates the question into a vector representation by the embedding process, which is then compared with the database to find the most relevant text (Query). Embedding is also designed to minimize interpretation errors, especially when the model encounters words with synonyms. This enables the system to understand the semantic relationships between words more accurately, producing relevant answers even if the keywords in the user's question differ slightly[10]. The found text is retrieved through the Data Retrieval process and then combined with the user's question to form a Prompt. These prompts provide additional context to a large language model (LLM Model) such as Mistral 7B, which uses its generative capabilities to generate answers. The final answer generated is designed to match the relevant information from the legal document and is presented to the user in an easy-to-understand format.

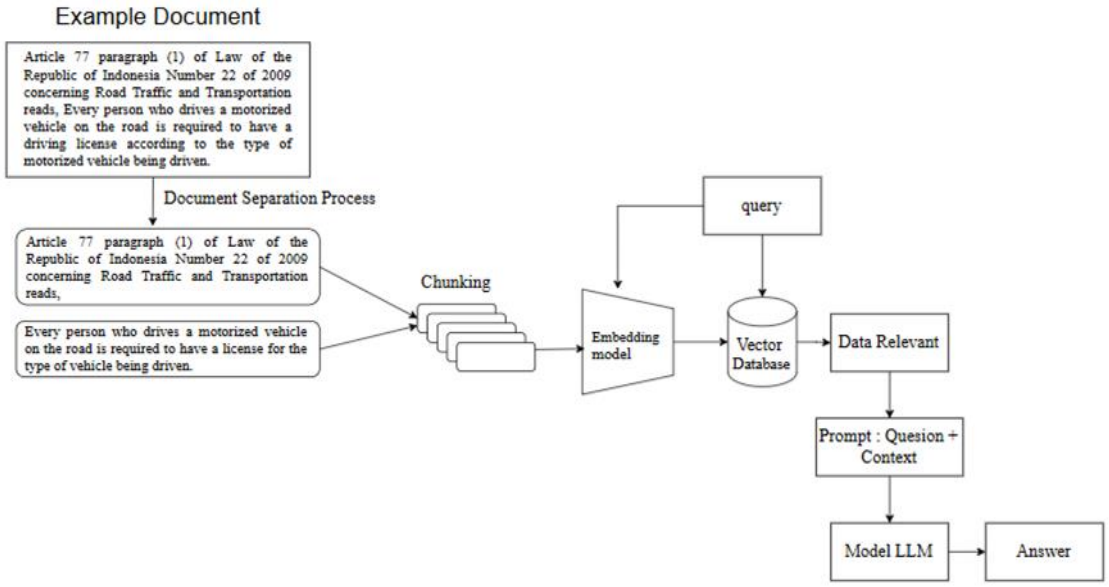

Figure 2. Flow of Retrieval-Augmented Generation (RAG) Method

2.4. Framework and Library

The framework used is Langchain. LangChain is a framework designed to build applications that use large language models (LLM) by facilitating developers in accessing additional data and interacting with other applications to support system development.[11], allowing the incorporation of various models and data sources for complex applications such as chatbots, recommendation systems, and others. The framework facilitates integration with various NLP tools and data stores. The libraries used in this project include Transformers, Accelerate, Bitsandbytes, Sentence-Transformers, Chromadb, LangChain-Community, and Deep Translator. Transformers from Hugging Face provides access to various large language models for natural language processing. Accelerate helps speed up the training and inference of large models by optimizing the distribution of computation on hardware such as GPUs. Bitsandbytes enables lower-precision floating-point to save memory and speed up training. Sentence-Transformers is a library designed to generate sentence representations with high efficiency. Chromadb is used for efficient database management and vector storage. LangChain-Community is a community extension for LangChain that provides various additional modules. Finally, Deep Translator is a library that automatically makes it easy to translate text using multiple online translation services.

2.5. Google Collaboratory

Google Collaboratory or Google Colab is a product of Google Research that allows users to write and run Python code through a browser and provides GPU computing with free access with restrictions or paid[12]. It does not require additional hardware configuration. This platform makes it easy to install libraries, load datasets, and use AI models because all computing is done on Google servers.

2.6. Gradio Library

This research also uses the Gradio library to facilitate the creation of the chatbot interface. Gradio is an open-source Python library that enables rapid user interface development for machine learning, data science, and web applications[13]. In the context of this research, Gradio is used to build a chatbot interface that users can use to ask questions related to traffic laws and get relevant answers.

2.7. BERTScore

The BERTScore method is used to assess the quality of the results. This method uses an NLP (Natural Language Processing) based text evaluation technique that uses the BERT (Bidirectional Encoder Representations from Transformers) model to assess the writing quality of a text and compares it with the reference text[14]. It calculates the similarity between two sentences by summing the cosine similarity of the embedding tokens of each sentence[15]. BERTScore produces a value between 0 and 1, where 1 indicates a perfect match between two texts, while 0 signifies a complete mismatch. The BERTScore calculation results in precision, recall, and f1-score, defined as the cosine similarity between the normalized contextual embeddings [16].

Recall (R_{BERT}) is calculated by matching the cosine similarity between tokens in the reference and result sentences. [17]. Embedding token xi in the candidate sentence with the most similar embedding token $\hat{x}$j in the reference sentence, where xi represents the embedding vector of the i-th token in the candidate sentence and $\hat{x}$j represents the embedding vector of the j-th token in the reference sentence. Then, it is averaged based on the total number of tokens in the candidate sentence$|x|$.

$$R_{BERT} = \frac{1}{|x|}\sum_{xi\in x} \max_{\hat{x}j\in\hat{x}} x_i{}^T \hat{x}_j \quad (1)$$

Precision (P_{BERT}) measures the accuracy of the predicted value in the result sentence compared to the reference sentence[18]. Precision is calculated by summing the maximum cosine similarity values between each embedding token $\hat{x}$j in the reference sentence with token embedding xi that is most similar in the candidate sentence, where $\hat{x}$j represents the embedding vector of the jth token in the reference sentence and xi represents the embedding vector of the i-th token in the candidate sentence, then the results are averaged based on the total number of tokens in the reference sentence$|\hat{x}|$.

$$P_{BERT} = \frac{1}{|\hat{x}|}\sum_{\hat{x}j\in\hat{x}} \max_{xi\in x} x_i{}^T \hat{x}_j \quad (2)$$

F1 Score(F_{BERT}) is an average comparison between Precision and Recall with weighting[19], which aims to balance the two metrics. This value measures how well the candidate text reflects the reference text as a whole[20], With (P_{BERT}) as a result of the Precision calculation and (RBERT) as a result of the Recall calculation, the candidate text is more similar to the reference text.

$$F_{BERT} = 2\,\frac{P_{BERT}\ .\ R_{BERT}}{P_{BERT} + R_{BERT}} \quad (3)$$

The BERTScore calculation starts by processing and comparing the model-generated answer with the reference answer. Each sentence in both answers is converted into an embedding representation using the BERT model. Next, the cosine similarity of each token in both sentences

is calculated, and the results are accumulated into a final score. A higher score indicates that the answer generated by the model is closer to the reference answer in terms of semantic similarity.

2.8. Fine-tuning

Fine-tuning the Mistral-7B language model began with installing and configuring the necessary libraries, including logging into the Hugging Face and Weights & Biases (W&B) platforms to track, visualize, and compare machine learning experiments. Libraries such as wandb, bitsandbytes, accelerate, peft, transformers, and trl are installed to support model training and more efficient memory usage through 4-bit quantization and Low-Rank Adaptation (LoRA). Once the environment is configured, users authenticate with Hugging Face and W&B accounts to integrate the model training process with real-time experiment tracking.

The next step is to load the relevant datasets from the CSV file and convert them into the Hugging Face format supported by the Transformers library using Dataset.from_pandas(). This Dataset trains the Mistral-7B-Instruct-v0.3 model with 4-bit quantization configured via BitsAndBytesConfig, which determines how data is processed and weights are updated during model training.[21], to reduce memory usage, Models and tokenizers are initialized with unique settings for cache and checkpointing to maximize training efficiency, LoRA (Low-Rank Adaptation) is applied to minimize memory usage and allow training on limited hardware by training smaller low-rank decomposition matrices. This technique injects a low-rank decomposition matrix into each layer of the Transformer, minimizing the parameters that need to be updated during training[22]. LoRA configuration in the code includes parameters such as lora_alpha, which controls the scale of low-rank matrix updates, lora_dropout to prevent overfitting through dropout, and r, which determines the rank size of the low-rank matrix to optimize training efficiency. The target module includes q_proj to compute queries, k_proj to generate keys, v_proj to create values, o_proj to project self-attention output, and gate_proj, which controls the flow of information in the model to ensure only relevant data is passed on.

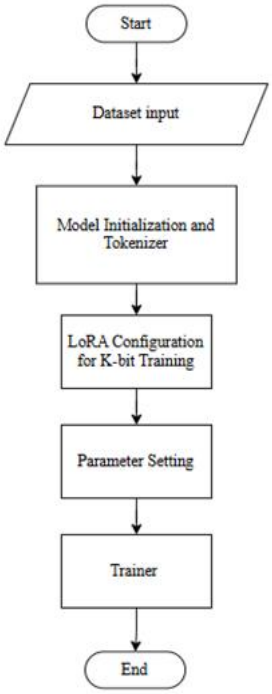

Figure 3. Fine-tuning

The training process uses SFTTrainer with parameter adjustments such as learning_rate = 2e-4, which sets how many weight updates are performed at each training iteration. This value was

chosen because it balances between stable convergence and training speed. A value that is too large can cause the model to fail to achieve convergence, while a value that is too small makes the training run slowly. The parameter per_device_train_batch_size = 4, which determines the amount of data processed per device and affects speed and memory usage, was chosen as the optimal midpoint for training the model. num_train_epochs = 1 ensures the entire dataset is used in training once. The paged_adamw_32bit optimizer was selected for memory efficiency, while weight_decay = 0.001 was applied as a regularization technique to prevent overfitting. The value 0.001 was chosen as it provides a sufficient level of light regulation to prevent overfitting without disturbing the already stable model weights; a higher weight decay could be too aggressive and change the weights significantly. The parameter max_grad_norm = 0.3 limits the maximum gradient value to maintain training stability and prevent exploding gradients, 0.3 was chosen because this value is safe enough to maintain training stability and is often used as a standard for training large models with sensitive optimizers such as AdamW. After training, the model was saved and uploaded to the Hugging Face Model Hub as “roif123/mistral_7b-instruct-UUD_NO_22_2009”. The entire process, from dataset input to training execution, is shown in Figure 3, which illustrates how each step is interconnected to maximize training efficiency, especially when using limited hardware.

3. Result and Discussion

3.1. Implementation of Results

After going through the system configuration process, Mistral, which has been integrated with the Retrieval-Augmented Generation (RAG) method, can answer various questions related to Law Number 22, Year 2009, as shown in Figure 4

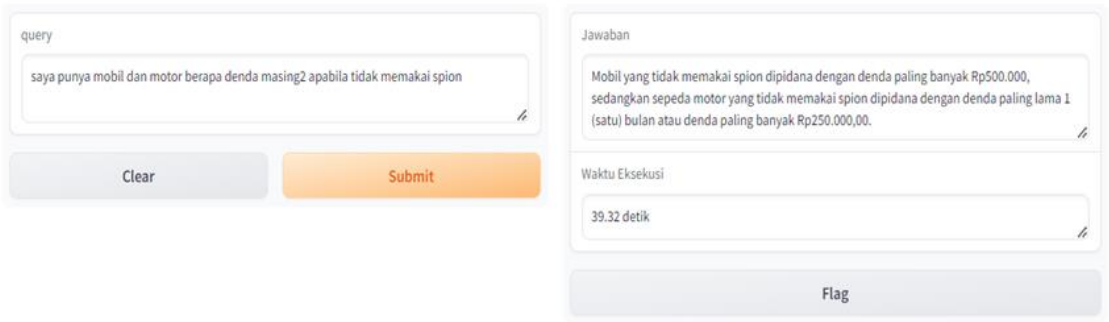

Figure 4. System Results

While the Mistral-7B model performed well in its output, it captured the answers from the legal documents provided. Still, some details were sometimes missed or not appropriately interpreted. Further fine-tuning is required to overcome these limitations and improve the precision and relevance of the results.

3.2. Comparison and Result

For comparison in Table 1, 10 example questions will be presented and calculated using BERTScore.

Table 1. Question

NO	Question
1	Apakah kalau saya tidak memakai spion pada sepeda motor akan terkena denda?
2	Apabila ada kendaraan berpapasan di persimpangan tanpa rambu lalu lintas, mana yang didahulukan?

NO	Question
3	Berbelok arah tanpa lampu sein dipidana dan di denda berapa?
4	Muatan yang melebihi batas apakah terkena denda?
5	Apabila terjadi kerusakan pada kendaraan dan tidak memasang segitiga pengaman, atau isyarat lain pada saat berhenti atau Parkir dalam keadaan darurat di Jalan,apakah diancam pidana?
6	Pasal uji berkala diwajibkan untuk jenis kendaraan apa saja?
7	Apakah saya wajib memakai helm di jalan raya?
8	Larangan merokok saat berkendara, dan dendanya?
9	Apa saja pengelompokkan kendaraan menurut kelas jalan?
10	Penyediaan perlengkapan Jalan diselenggarakan oleh?

Comparisons were made using three models, namely Mistral 7B, Mistral 7B fine-tuning, and Chat GPT-4.o; the selection of Chat GPT 4.o for external model comparison because ChatGPT is a natural language-based AI system that provides answers and solutions based on existing data.[23]This model is often used and accessed through the website https://chatgpt.com/. Here, experiments were carried out by creating a new chat, uploading a dataset, and providing prompts based on table data to generate answers. The BERTScore calculation results for each model can be seen in Tables 2, 3, and 4.

Table 2 shows the BERTScore results for the Mistral 7B model, with Precision, Recall, and F1 Score values obtained from 10 questions.

Table 2. BERTScore Mistral 7B

Question	Precision	Recall	F1
1	0.9220	0.8431	0.8808
2	0.9016	0.8575	0.8790
3	0.8528	0.7918	0.8212
4	0.9121	0.8234	0.8655
5	0.8844	0.8928	0.8886
6	0.9304	0.9546	0.9423
7	0.8746	0.8247	0.8489
8	0.9620	0.9139	0.9373
9	0.9026	0.8573	0.8794
10	0.8383	0.8842	0.8606

In Table 3, the Mistral 7B model results after the fine-tuning process show improvements in several Precision, Recall, and F1 Score metrics compared to the model before fine-tuning. Although there are some lower metric results, the Precision, Recall, and F1 Score performance has improved, as shown in Table 5.

Table 3. BERTScore Mistral 7B Fine-tuning

Question	Precision	Recall	F1
1	0.9091	0.8442	0.8755
2	0.9211	0.8378	0.8774

Question	Precision	Recall	F1
3	0.8596	0.7878	0.8221
4	0.9460	0.9074	0.9263
5	1.0000	1.0000	1.0000
6	0.9942	0.9952	0.9947
7	0.8799	0.7936	0.8345
8	0.9625	0.9146	0.9380
9	0.9509	0.9124	0.9313
10	0.9558	0.9465	0.9511

Meanwhile, Table 4 presents the BERTScore results for Chat GPT-4.0, which is used as an external model for comparison.

Table 4. BERTScore ChatGPT

Question	Precision	Recall	F1
1	0.8635	0.8700	0.8667
2	0.8505	0.8372	0.8438
3	0.8533	0.8035	0.8277
4	0.8499	0.8093	0.8291
5	0.8538	0.8724	0.8630
6	0.8087	0.8452	0.8265
7	0.8751	0.8872	0.8811
8	0.8093	0.8558	0.8319
9	0.7855	0.7873	0.7864
10	0.7607	0.8605	0.8075

Table 5 summarizes each model's average F1 Score values to provide an overall picture of their performance.

Table 5. F1 Score Average

Models	F1 Score Average
Mistral 7B	0.8804
Mistral 7B Fine-tuning	0.9151
Chat GPT 4.o	0.8364

The F1 score results obtained show a significant increase after the fine-tuning process. The F1 score was 0.88036 for the Mistral 7B model, but after fine-tuning, this score increased to 0.9151. This difference of 0.03474 indicates that fine-tuning has improved the accuracy and performance of the model in understanding and producing more correct answers. This improvement shows that the optimized model can capture better context in the data, resulting in more relevant and accurate answers compared to the model version before fine-tuning.

However, the average F1 score obtained by Chat GPT is 0.8364. Although this score is not very high, the answers produced by the model appear descriptively better. It is important to note that BERTScore may give lower scores because it evaluates text similarity differently than human perception. BERTScore compares each token using a vector representation, so small word choice or sentence structure changes, such as synonyms, can lower the score.[20]. This method calculates semantic similarity in more detail, and although the meaning of the text remains the same, these variations affect the Precision, Recall, and F1 scores, making them appear lower.

In addition to finding accurate answers, the quality of the questions asked greatly affects the results obtained. For example, more specific questions such as: “Pasal berapa Uji berkala

diwajibkan untuk jenis kendaraan apa saja?" contain important key elements, namely 'Pasal,' 'Uji Berkala,' and "Jenis Kendaraan." These key elements help narrow down the scope of the information search, as illustrated in Table 6 with the following example.

Table 6. Keyword Implementation

Question	Answer Results	Reference
Pasal berapa Uji berkala diwajibkan untuk jenis kendaraan apa saja?	Uji berkala diwajibkan untuk jenis kendaraan seperti mobil penumpang umum, mobil bus, mobil barang, kereta gandengan, dan kereta tempelan yang dioperasikan di jalan. Hal ini disebutkan dalam Pasal 53 Undang-Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan.	Uji berkala diwajibkan untuk jenis kendaraan seperti mobil penumpang umum, mobil bus, mobil barang, kereta gandengan, dan kereta tempelan yang dioperasikan di jalan. Hal ini disebutkan dalam Pasal 53 Undang-Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan.

With the specification in the question, the model or system can focus more on searching and analyzing relevant documents to produce appropriate answers based on the context. This specification helps narrow the scope of the information search, allowing the model to access and process the most relevant data more efficiently. Therefore, clarity and precision in formulating the question become crucial factors in achieving the desired answer accuracy, ensuring that the results are accurate and relevant to the specific context asked.

4. Conclusion

This research has successfully demonstrated that combining the Mistral-7B language model with the Retrieval-Augmented Generation (RAG) method can significantly improve legal data processing, especially in Law Number 22 Year 2009 on Road Traffic and Transportation. The fine-tuning process performed on this model significantly improved the performance of the model, with an increase in F1 score from 0.88036 to 0.9151. In comparison, the GPT-4.0 model has an F1 score of 0.8364. The difference in improvement of 0.03474 in Mistral-7B and 0.0348 in GPT-4.0 indicates that fine-tuning has successfully improved the accuracy and performance of the model in understanding and producing more precise answers. This improvement means the optimized model can capture better context, resulting in more relevant and accurate answers than the model version before fine-tuning.

In addition, this research emphasizes the importance of specification in the questions asked to achieve higher accuracy in answering them. The model or system can focus more on searching and analyzing relevant documents by using key elements in the question, resulting in more precise answers. The results of this study are expected to serve as a basis for further development in the integration of AI technology in the legal system in Indonesia, as well as provide a practical guide for the application of AI in other legal contexts.

References

[1] M. N. Setiawan, R. S. Roring, Y. D. Atma, and H. Tetiawadi, "Studi Empiris Terhadap Asistensi Artificial Intelligence (AI) Dalam Rancang Bangun Aplikasi," *Digital Transformation Technology (Digitech)*, vol. 4, no. 1, pp. 364–373, Jun. 2024, doi: 10.47709/digitech.v4i1.4115.

[2] E. N. Ravizki and Lintang Yudhantaka, "Artificial Intelligence Sebagai Subjek Hukum: Tinjauan Konseptual dan Tantangan Pengaturan di Indonesia," *Notaire*, vol. 5, no. 3, pp. 351–376, Oct. 2022, doi: 10.20473/ntr.v5i3.39063.

[3] A. Q. Jiang *et al.*, “Mistral 7B,” Oct. 2023, [Online]. Available: https://arxiv.org/abs/2310.06825
[4] M. Salıcı and Ü. E. Ölçer, “Impact of Transformer-Based Models in NLP: An In-Depth Study on BERT and GPT,” in *2024 8th International Artificial Intelligence and Data Processing Symposium (IDAP)*, IEEE, Sep. 2024, pp. 1–6. doi: 10.1109/IDAP64064.2024.10710796.
[5] A. Salemi and H. Zamani, “Evaluating Retrieval Quality in Retrieval-Augmented Generation,” in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA: ACM, Jul. 2024, pp. 2395–2400. doi: 10.1145/3626772.3657957.
[6] I. Pujiono, I. M. Agtyaputra, and Y. Ruldeviyani, “Implementing Retrieval-Augmented Generation And Vector Databases For Chatbots In Public Services Agencies Context,” *JITK (Jurnal Ilmu Pengetah. dan Teknol. Komputer)*, vol. 10, no. 1, pp. 216–223, Aug. 2024, doi: 10.33480/jitk.v10i1.5572.
[7] K. M. Fitria, “Information Retrieval Performance in Text Generation using Knowledge from Generative Pre-trained Transformer (GPT-3),” *Jambura Journal of Mathematics*, vol. 5, no. 2, pp. 327–338, Aug. 2023, doi: 10.34312/jjom.v5i2.20574.
[8] A. T. Laksana, S. Sylviani, and A. Triska, “Studi Penerapan Konsep Vektor Dalam Permasalahan Penyisipan Kata-Kata Melalui Proses Normalisasi Vector Dan Transformasi Orthogonal,” *Jurnal Lebesgue Jurnal Ilmiah Pendidikan Matematika, Matematika dan Statistik*, vol. 5, no. 2, pp. 634–641, Aug. 2024, doi: 10.46306/lb.v5i2.493.
[9] F. Almeida and G. Xexéo, “Word Embeddings: A Survey,” Jan. 2019, doi: https://doi.org/10.48550/arXiv.1901.09069.
[10] K. Muludi, K. M. Fitria, J. Triloka, and S. -, “Retrieval-Augmented Generation Approach: Document Question Answering using Large Language Model,” *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 15, no. 3, 2024, doi: 10.14569/IJACSA.2024.0150379.
[11] S. Rahayu, N. S. Harahap, S. Agustian, and Pizaini, “Penerapan Teknologi LangChain pada Question Answering System Fikih Empat Madzhab,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, pp. 974–983, Jul. 2024, doi: https://doi.org/10.57152/malcom.v4i3.1397.
[12] G. I. E. Soen, Marlina, and Renny, “Implementasi Cloud Computing dengan Google Colaboratory pada Aplikasi Pengolah Data Zoom Participants,” *JITU Journal of Informatic Technology and Communication*, vol. 6, no. 1, pp. 24–30, Jun. 2022, doi: 10.36596/jitu.v6i1.781.
[13] Gradio, “Gradio: Build Machine Learning Web Apps in Python.” Accessed: Sep. 12, 2024. [Online]. Available: https://pypi.org/project/gradio
[14] C. S. R. Chan, C. Pethe, and S. Skiena, “Natural language processing versus rule-based text analysis: Comparing BERT score and readability indices to predict crowdfunding outcomes,” *J. Bus. Ventur. Insights*, vol. 16, p. e00276, Nov. 2021, doi: 10.1016/j.jbvi.2021.e00276.
[15] M. I. Syah, N. S. Harahap, Novriyanto, and S. Sanjaya, “PENERAPAN RETRIEVAL AUGEMENTED GENERATION MENGGUNAKAN LANGCHAIN DALAM PENGEMBANGAN SISTEM TANYA JAWAB HADIS BERBASIS WEB,” *ZONAsi : Jurnal Sistem informasi*, vol. 6, no. 2, pp. 370–379, May 2024, doi: 10.31849/zn.v6i2.19940.
[16] I. J. Unanue, J. Parnell, and M. Piccardi, “BERTTune: Fine-Tuning Neural Machine Translation with BERTScore,” Jun. 2021, [Online]. Available: https://arxiv.org/pdf/2106.02208
[17] A. T. U. B. Lubis, N. S. Harahap, S. Agustian, M. Irsyad, and I. Afrianty, “Question Answering System pada Chatbot Telegram Menggunakan Large Language Models (LLM) dan Langchain (Studi Kasus UU Kesehatan),” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, pp. 955–964, May 2024, doi: 10.57152/malcom.v4i3.1378.
[18] F. Fajri, B. Tutuko, and S. Sukemi, “Membandingkan Nilai Akurasi BERT dan DistilBERT pada Dataset Twitter,” *JUSIFO (Jurnal Sistem Informasi)*, vol. 8, no. 2, pp. 71–80, Dec. 2022, doi: 10.19109/jusifo.v8i2.13885.
[19] P. L. Romadloni, B. A. Kusuma, and W. M. Baihaqi, “KOMPARASI METODE PEMBELAJARAN MESIN UNTUK IMPLEMENTASI PENGAMBILAN KEPUTUSAN DALAM MENENTUKAN PROMOSI JABATAN KARYAWAN,” *JATI : Jurnal Mahasiswa Teknik*

Informatika, vol. 6, no. 2, pp. 622–628, Sep. 2022, doi: 10.36040/jati.v6i2.5238.
[20] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," Apr. 2019, [Online]. Available: http://arxiv.org/abs/1904.09675
[21] K. Pandya, "PEFT-MedAware: Large Language Model for Medical Awareness," Nov. 2023, [Online]. Available: https://arxiv.org/abs/2311.10697
[22] W. Xia, C. Qin, and E. Hazan, "Chain of LoRA: Efficient Fine-tuning of Language Models via Residual Learning," Jan. 2024, [Online]. Available: https://arxiv.org/abs/2401.04151
[23] R. Darman, "Peran ChatGPT Sebagai Artificial Intelligence Dalam Menyelesaikan Masalah Pertanahan dengan Metode Studi Kasus dan Black Box Testing," *Tunas Agraria*, vol. 7, no. 1, pp. 18–46, Jan. 2024, doi: 10.31292/jta.v7i1.256.