Implementation of a transformer-based question answering model in KutubBot for the Kutubut Tis'ah Hadith Corpus

Ana Tsalitsatun Ni'mah[1*], *Ahmad* Yahya Surya Winata[2], *Wanda* Ramansyah[1], *Sulton* Nawawi[1], *Riva* Ananada Yuanova Kamajaya[1], *Adam* Muhammad Mansur[1], and *Adamu* Muhammed[3]

[1]Informatics Education, Universitas Trunodjoyo Madura, Jl. Raya Telang, Telang, Kamal District, Bangkalan Regency, East Java 69162, Indonesia
[2]Management, Universitas Trunodjoyo Madura, Jl. Raya Telang, Telang, Kamal District, Bangkalan Regency, East Java 69162, Indonesia
[3]Deputy Director of Social Management Ardo-Kola LGA, Nigeria

Abstract. Current technological developments, particularly in the fields of Artificial Intelligence and Natural Language Processing (NLP), have significantly improved question-answering systems. This technology can be utilized in the fields of education and religion, particularly in the study of hadith. One challenge in hadith learning is the large number of reference sources contained in the Kutubut Tis'ah (nine main hadith books), making it time-consuming for users to find relevant answers. Therefore, a system capable of providing fast, accurate, and appropriate answers based on the hadith text is needed. This research aims to develop an artificial intelligence-based chatbot, called kutubBot, to answer questions related to Kutubut Tis'ah. This system employs a Transformer-based Question Answering approach to comprehend user questions and provide relevant answers based on the hadith corpus. The research stages included data collection from nine hadith books, text preprocessing, model training, and performance evaluation using accuracy, precision, recall, and F1-score metrics. Test results showed that KutubBot achieved good results, with an 97.8% accuracy, 90.3% average precision, 93.45% recall, and an F1-score of 91.86%. These evaluation values indicate that the system has a high level of accuracy and completeness in providing answers that are contextually relevant to the hadith.

1 Introduction

The rapid development of information technology has transformed how people access information. The abundance of digital data and documents across various media is now overwhelming, requiring users to rely on systems that provide information quickly, accurately, and tailored to their needs [1][2]. One innovation emerging from this need is the Question Answering Chatbot (QA Chatbot) system [3]. This type of chatbot is designed to answer user questions using artificial intelligence automatically [4][5].

The Question Answering Chatbot system is a development of the fields of Information Retrieval and Natural Language Processing (NLP) [6][7]. The system not only searches for relevant documents but also understands the user's intent and generates contextually relevant answers [8]. One type of document that is particularly appealing as a basis for developing a QA system is the Hadith [9][10].

The Hadith is the second source of Islamic law after the Quran, containing the sayings, deeds, and decrees of the Prophet Muhammad (peace be upon him) [11]. Each Hadith consists of a sanad (chain of narrators) and matan (content) [12]. The large number of hadith books, each with its own categories, structure, and language, makes searching for and studying Hadith a challenge for the general public [13]. To simplify this process, a question-answering chatbot is needed that can provide direct answers to questions about Hadith, rather than simply displaying hadith texts containing specific keywords [13].

Previous research in the field of digital Hadith has generally focused on hadith search systems using term-weighted information retrieval approaches such as Term Frequency–Inverse Document Frequency (TF-IDF) and its developments, such as TF-IDF–Inverse Class Frequency (TF-IDF–ICF) and TF-IDF–Inverse Class Space Density Frequency (TF-IDF–ICS$_\delta$F) [14][15]. This approach calculates the weight of each word and determines the similarity between the query and the document [16]. While effective at finding relevant text, this approach has limitations because it can only perform keyword matching, not understand the context of the question or the semantic meaning of the sentence [17].

Recent developments in NLP have introduced transformer models, such as BERT (Bidirectional Encoder Representations from Transformers), RoBERTa, DistilBERT, and GPT, which can deeply understand sentence context through attention mechanisms [18][19]. Transformer-based models have

* Corresponding author: ana.tsalits@trunojoyo.ac.id

proven superior in various natural language understanding tasks, including question answering [20]. Several previous studies have demonstrated the superiority of this approach [21]. BERT successfully improved accuracy in QA tasks on the SQuAD (Stanford Question Answering Dataset) dataset [22][23]. Furthermore, research applying transformer models to religious texts has shown that these models can better understand semantic context than keyword-based methods [24]. In Indonesia, a BERT-based Islamic chatbot is being developed to answer questions about the Quran and Hadith with high accuracy [25].

Based on these studies, this study proposes the development of a transformer model-based Question Answering Chatbot to answer questions about hadith, called KutuBot (Kutubut Tis'ah Chatbot). This chatbot is designed to understand the user's natural language and provide answers sourced from nine main hadith books (Kutubut Tis'ah). This system works by using a pre-trained transformer model, which is then fine-tuned on a hadith dataset that has undergone preprocessing steps, such as text cleaning, tokenization, lemmatization, and annotation, to strengthen understanding of the hadith context.

Unlike conventional hadith search systems, the transformer-based Question Answering Chatbot not only displays hadith containing question keywords but is also capable of extracting specific answers from the hadith text based on semantic understanding. This system is integrated into a web-based platform that enables users to ask questions in natural language and receive relevant, concise, and contextually relevant answers.

With this approach, the transformer-based Question Answering Chatbot is expected to improve the efficiency of searches for hadith information and support digital-based Islamic learning. Furthermore, this research contributes to the development of Natural Language Processing (NLP) technology for the Indonesian language, particularly for religious texts with complex structures and meanings.

2 Method

2.1 Research approach

This research uses a quantitative experimental approach based on systems development research (SDR) to build and test the performance of the KutuBot chatbot, which can answer hadith questions from the Kutubut Tis'ah book using a transformer-based question answering approach.

This research focuses on developing an Inverse Hadith Space Density Frequency ($IHS_\delta F$) term weighting algorithm, integrated into a semantic retrieval system, to increase the relevance of hadith answers before the transformer model processes them.

2.2 Research data

The dataset used consists of 63,000 question-answer pairs from nine main hadith books (Kutubut Tis'ah), namely:
- Sahih Bukhari
- Sahih Muslim
- Sunan Abu Dawud
- Sunan Tirmidhi
- Sunan Nasa'i
- Sunan Ibn Majah
- Musnad Ahmad
- Muwatta Malik
- Sunan Darimi

Each hadith is accompanied by metadata, including book name, chapter, and thematic category, to support the space-based term weighting process.

2.3 Research stages

The research stages included:
- Data collection and cleaning (data collection & preprocessing)
- Designing the $IHS_\delta F$ term weighting algorithm
- Implementing a transformer-based question answering system
- Testing and evaluating system performance

The general flowchart of the research can be described as follows:
Question Input → Preprocessing → $IHS_\delta F$ Weighting → Relevant Hadith Selection → Transformer QA Model → Answers.

2.4 Data preprocessing

Before weighting, the hadith texts and user queries undergo several pre-processing stages:
- Case Folding: Converting the entire text to lowercase.
- Tokenization: Breaking the text into individual words.
- Stopword Removal: Removing common words that carry no semantic significance.
- Stemming/Lemmatization: Returning words to their base form.
- Normalization: Standardizing Arabic-Latin transliteration (e.g., prayer → prayer).

The result of this stage is a clean corpus of hadith texts ready for term weighting.

2.5 $ICS_\delta F$ and $IHS_\delta F$ term weighting concept

2.5.1 Term Frequency (TF)

Term frequency (TF) assumes each term has a proportional importance to the number of times it appears in the document [26]. Term frequency (TF) is the simplest method for weighting terms. Term frequency is the frequency with which a term appears in the document in question [27]. The greater the number of times a term appears (higher TF) in the document, the greater its weight, or the greater its suitability value [28]. Term Frequency (TF) has several formulas that can be used:

- Binary TF, which only considers whether a term is present in a document. If present, it is given a value of one (1); otherwise, it is given a value of zero (0).
- Raw TF, which assigns a TF value based on the number of occurrences of a term in the document. For example, if a term appears five (5) times, it has a value of five (5).
- Logarithmic TF, which avoids the dominance of documents containing few terms in the query but with high frequency. Logarithmic TF is visualized in the equation (1).

$$TF = \begin{cases} 1 + log_{10}\left(f_{t,d}\right), & f_{t,d} > 0 \\ 0, & f_{t,d} = 0 \end{cases} \quad (1)$$

where the value of $f_{t,d}$ is the frequency of term *(t)* in document *(d)*. So if a term or terms are present in a document 5 times, then the weight obtained is 1 + log(5) = 1.699. But if the term is not present in the document, the weight is zero (0).

2.5.2 Inverse Document Frequency (IDF)

Inverse Document Frequency (IDF) considers the occurrence of terms in a document collection [29]. The IDF function assigns the lowest score to terms that appear in many documents in the document space $D = d_1, d_2, \ldots, d_n$. The background to this weighting is that terms that rarely appear in a document collection are highly valuable [30]. The importance of each term is assumed to be inversely proportional to the number of documents containing the term [31]. IDF is visualized in Equations (2) and (3).

$$df_{(t_i)} = \sum d_{(t_i)} \quad (2)$$

where $df_{(t_i)}$ is the frequency of documents containing term *i*, $d_{(t_i)}$is the document containing term *i*.

$$idf = \log \frac{D}{df_{(t_i)}} \quad (3)$$

where, *idf* is the inverse of document frequency $(df_{(t_i)})$, *D* is the total number of documents.
Then *IDF* combined with TF is visualized in equation (4).

$$W_{TF*IDF(t_i,d_j)} = tf_{t_i,d_j} \times \left(log \frac{D}{df_{(t_i)}}\right) \quad (4)$$

where, $W_{TF*IDF(t_i,d_j)}$ is the weight of term *i* in document *j*, $tf_{t_i,d_j}$is the number of term *i* in document *j*, *D* is the total number of documents, and $df_{(t_i)}$ is the total number of documents containing term *i*.

2.5.3 Inverse Class Frequency (ICF)

Inverse Class Frequency (ICF) is adopted from the IDF method, namely by using the inverse ratio of the number of categories to the number of categories containing the term. In the case of class-oriented indexing, a subset of documents from the document space $D = d_1, d_2, \ldots, d_n$ is allocated to a particular category. So the more frequently a term appears in documents in that category, the closer the term's weight is to 0. The ICF function provides the lowest score for terms that appear in several categories in the category space $C = C_1, C_2, \ldots, C_n$. The equation of the ICF can be seen in Equations (5) and (6).

$$cf_{(t_i)} = \sum c_{(t_i)} \quad (5)$$

where, $cf_{(t_i)}$ is the frequency of the category containing term *i*, $c_{(t_i)}$ is the category containing term *i*.

$$icf = \log \frac{C}{cf_{(t_i)}} \quad (6)$$

where, icf is the inverse of class frequency $cf_{(t_i)}$, *C* is the total number of categories.

Therefore, the numerical representation of a term is the product of Term Frequency (local parameter), IDF (global parameter), and ICF (global category parameter). The TF x IDF x ICF equation is visualized in equation (7).

$$W_{TF*IDF*ICF(t_i,d_j,c_k)} = tf_{t_i,d_j} \times \left(\log \frac{D}{df_{(t_i)}}\right) \times \left(\log \frac{C}{cf_{(t_i)}}\right) \quad (7)$$

where, $W_{TF*IDF*ICF(t_i,d_j,c_k)}$ is the weight of term *i* in document *j*, $tf_{t_i,d_j}$is the number of term *i* in document *j*, *D* is the total number of documents, $df_{(t_i)}$is the total number of documents containing term *i*, *C* is the total number of categories, $cf_{(t_i)}$is the total number of categories containing term *i*.

2.5.4 Inverse Class Space Density Frequency ($ICS_\delta F$)

$ICS_\delta F$ calculates document density in the category space for each term. Since the ICF assigns the lowest score to terms that appear in multiple categories, without considering the category space, the $ICS_\delta F$ calculation is proposed. This term weighting method is to improve the classification performance. These identical documents associated with a certain term may be a subset of a certain category c_k. Therefore, it is important to explore the characteristics of the term occurrence in the document space $D = \{d_1, d_2, \ldots, d_n\}$ and the category space. In category-oriented indexing, a subset of documents from the global document space is assigned to categories c_k ($k = 1, 2, \ldots, m$) based on their topics. Therefore, the category space is defined as $C = \{(d_{11}, d_{12}, \ldots, d_{1n})$ *member of* C_1, $(d_{21}, d_{22}, \ldots, d_{2n})$ *member of* $C_2, \ldots, (d_{m1}, d_{m2}, \ldots, d_{mn})$ *member of* $C_m\}$. ICSδF begins by calculating the class density (C_δ), namely, calculating documents containing terms in a certain category $c_k$visualized in equation (8).

$$C_\delta(t_i) = \frac{n_{c_k}(t_i)}{N_{c_k}} \quad (8)$$

where, $C_\delta(t_i)$ is the category density (class density) for term *i,* $n_{c_k}(t_i)$ is the number of documents in category c_k that contain term i, N_{c_k} is the total number of documents in category c_k.
Then, we continue by calculating the category space density, which is the sum of the densities of all existing categories (CS_δ), visualized in equation (9).

$$CS_\delta(t_i) = \sum_{c_k} C_\delta(t_i) \tag{9}$$

where, $CS_\delta(t_i)$ is the class space density for term *i*, $C_\delta(t_i)$ is the category density for term *i*, $c_k$is the category (k = *1, 2, ..., m*).
Then the result of the category space density ($CS_\delta(t_i)$) is inverse according to the previous concept in TF-IDF-ICF, visualized in equation (10).

$$ICS_\delta F(t_i) = log\left(\frac{C}{CS_\delta(t_i)}\right) \tag{10}$$

where, $ICS_\delta F(t_i)$ is the inverse class space density frequency for term *i*, *C* is the total number of categories, $CS_\delta(t_i)$ is the category space density for term *i*.
The next step is to multiply the result of the inverse category space density for term *i* ($ICS_\delta F(t_i)$) by TF-IDF, as in equation (11).

$$W_{TF*IDF*ICS_\delta F(t_i,d_j,c_k)} = tf_{t_i,d_j} \times \left(log\frac{D}{df_{(t_i)}}\right) \times \left(log\frac{C}{CS_\delta(t_i)}\right) \tag{11}$$

where, $W_{TF*IDF*ICS_\delta F(t_i,d_j,c_k)}$ is the weight of term *i* in document *j* in category *k*, tf_{t_i,d_j} is the number of term *i* in document *j*, *D* is the total number of documents, $df_{(t_i)}$ is the number of documents containing term *i*, *C* is the total number of categories, $CS_\delta(t_i)$ is the density of the category space with respect to term *i*.

2.5.5 IHSδF (Inverse Hadith Space Density Frequency)

This research proposes a development of $ICS_\delta F$, namely Inverse Hadith Space Density Frequency ($IHS_\delta F$). While $ICS_\delta F$ considers the spatial density of categories relative to a term, $IHS_\delta F$ focuses more on the spatial density of Hadith books relative to a term. The spatial density of Hadith books is calculated to determine the weight of a term when the rarity of that term is also taken into account in the total number of Hadith books. The less frequently a term appears in many Hadith books, the higher its inverse value. The first step in calculating IHSδF is to calculate the density of hadith, as in equation (12).

$$H_\delta(t_i) = \frac{n_{h_l}(t_i)}{N_{h_l}} \tag{12}$$

where, $H_\delta(t_i)$ is the density of the book with respect to term *i*, $n_{h_l}(t_i)$ is the number of hadiths in book *l* containing term *i*, N_{h_l} is the total number of hadiths in book *l*, and h_l is the book (*l = 1, 2, ..., 9*).
The density of categories (C_δ) and the density of Hadith Books (H_δ) are very different, C_δ calculates how much the density/occurrence of a term in a particular category by comparing the number of documents that have the term for a specific category compared to the total number of records in a specific category while H_δ calculates how much the density/occurrence of a term in a specific Hadith Book by comparing the number of documents that have the term in a specific Hadith book compared to the total number of records in a specific Hadith book. The density of hadiths with respect to term *i* is summed to obtain the spatial density of the Hadith book with respect to term *i*, as in equation (13).

$$HS_\delta(t_i) = \sum_{h_l} H_\delta(t_i) \tag{13}$$

where, $HS_\delta(t_i)$ is the space density of hadith books for term *i*, $H_\delta(t_i)$ is the density of hadith books for term *i*, and $h_l$is the 9 hadith books (*l = 1, 2, ..., 9*).
Classroom density (CS_δ) differs from Hadith space density (HS_δ). CS_δ measures the frequency of occurrence of a term across all classes, while HS_δ measures the frequency of occurrence of a term across all Hadith books.
Next, the Hadith space density result is inversely calculated for term *i* to determine the degree of rarity of the term within the Hadith space, as shown in equation (14).

$$IHS_\delta F(t_i) = log\left(\frac{H}{HS_\delta(t_i)}\right) \tag{14}$$

where, $IHS_\delta F(t_i)$ is the inverse of the book space density with respect to term *i*, *H* is the total number of books (9), and $HS_\delta(t_i)$ is the book space density with respect to term *i*.
Next, the inverse result is multiplied by $TF * IDF * IHS_\delta F$ to determine the term weight that takes into account the class space density and also the hadith space density as in equation (15).

$$W_{TF*IDF*ICS_\delta F*IHS_\delta F(t_i,d_j,c_k,h_l)} = tf_{t_i,d_j} \times \left(log\frac{D}{df_{(t_i)}}\right) \times \left(log\frac{C}{CS_\delta(t_i)}\right) \times \left(log\left(\frac{H}{HS_\delta(t_i)}\right)\right) \tag{15}$$

where, $W_{TF*IDF*ICS_\delta F*IHS_\delta F(t_i,d_j,c_k,h_l)}$ is the weight of term *i* in hadith *j* in category *k* in book *l*, tf_{t_i,d_j} is the number of term *i* in hadith *j*, *D* is the total number of hadiths, $df_{(t_i)}$ is the number of hadiths containing term *i*, *C* is the total number of categories, $CS_\delta(t_i)$ is the density of category space for term *i*, *H* is the total number of books (9), and $HS_\delta(t_i)$ is the density of category space for term *i*. The results of term weighting by considering the density of category space and density of hadith space give higher values to rare terms in categories and

hadiths, and give low values to terms that appear frequently.

2.5.6 Integration of IHSδF in QA system

The IHSδF weighting results are used to select the most relevant hadith candidates for the user's question.
Only the Top-N hadiths with the highest scores will proceed to the next stage: the transformer-based question-answering model (IndoBERT fine-tuned).

2.6 Transformer-based question answering model

A transformer-based Question Answering model was used to extract answers from hadith texts filtered using IHSδF.
The model used was IndoBERT-base with the following configuration:
- Batch size: 16
- Learning rate: 5e-5
- Epochs: 5
- Optimizer: AdamW
- Loss Function: Cross-Entropy
- Validation Split: 10%

This model was trained using data pairs (question, hadith text, answer) to develop the ability to understand semantic context and determine the text segment that best answers the question.

2.7 System evaluation

The evaluation was conducted using two approaches:
- Quantitative Evaluation:
  Metrics used included Accuracy, Precision, Recall, and F1-Score to measure the system's performance in answering questions correctly.
- Qualitative Evaluation:
  Three hadith experts assessed the suitability of the meaning and accuracy of the chatbot's responses to the hadith sources.

3 Results and discussion

3.1 System Evaluation

The KutubBot chatbot system was developed using a two-stage architecture:
- Hadith Retrieval Stage:
  Uses IHSδF weighting to determine the most relevant hadith to the user's question.
- Answer Extraction Stage:
  Uses a transformer-based QA model (fine-tuned IndoBERT) to extract textual answers from selected hadith.

This process was implemented in Python 3.11 with the following core libraries:
- Transformers (HuggingFace) for IndoBERT modeling
- FAISS for fast Top-N document search
- Scikit-learn for performance evaluation

The system was tested using 63,000 translated Indonesian hadith data sets, consisting of 54,000 training data sets and 9,000 test data sets.

3.2 Term weighting results with IHSδF

The term weighting process produces a value that reflects the word's level of distinctiveness within the context of the category and the hadith book.
For example, the term "intention" has a high weight in the worship category in Sahih Bukhari but a low weight in Musnad Ahmad, due to its higher frequency in that book. Table 1 compares the average weights of several terms calculated using ICSδF and IHSδF.

Table 1. Average ICSδF and IHSδF.

Term	ICSδF	IHSδF	Perubahan Bobot (%)
intention	0.254	0.392	+54.3%
zakat	0.317	0.426	+34.4%
jihad	0.281	0.401	+42.7%
buy and sell	0.239	0.358	+49.8%
fast	0.266	0.377	+41.7%

The increased weighting of the IHSδF method indicates that the model places greater emphasis on words specific to the book's context, rather than simply on thematic categories. This is important because hadith discussing the same topic in different books often use varying terms.

3.3 Document selection results (Retrieval stage)

The system was tested to select the Top 10 most relevant hadiths from 1,000 random test questions.
Table 2 shows the success rate (recall @ 10) in finding relevant hadiths.

Table 2. Success Rate (Recall@10).

Weighting Method	Recall@10	Precision@10	Mean Time (ms)
TF-IDF	0.81	0.78	14.3
ICSδF	0.87	0.84	17.9
IHSδF	**0.93**	**0.89**	**19.2**

The IHSδF method provided a +6% recall improvement compared to ICSδF and +12% compared to TF-IDF. This indicates that the added "hadith space" dimension improved the relevance of the hadiths found prior to the answer extraction process.

3.4 Question answering stage results

The second stage involved fine-tuning the IndoBERT QA model using the Top-N results from the IHSδF stage. The model was tested on 9,000 test datasets (63,000 total datasets × 15%). Table 3 shows the quantitative evaluation results.

Table 3. Success Rate (Recall@10).

Retrieval Method + QA	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
TF-IDF + Transformer QA	89.42	82.33	83.10	82.71
$ICS_{\delta}F$ + Transformer QA	94.05	86.90	90.12	88.48
$IHS_{\delta}F$ + Transformer QA	97.81	90.32	93.45	91.86

The combination of $IHS_{\delta}F$ + Transformer QA yielded significant improvements across all metrics compared to the previous method. Accuracy increased by 3.76% over $ICS_{\delta}F$ and 8.39% over TF-IDF. The highest Recall and F1-Score demonstrate the system's ability to find and extract the correct answers according to the context of the hadith. This improvement is due to $IHS_{\delta}F$'s more selective weighting of important terms in each hadith book, resulting in more relevant and contextual documents passed to the QA model.

3.5 Qualitative evaluation

Three hadith experts assessed 100 random answers from the chatbot based on three aspects:
- The relevance of the hadith's meaning to the question.
- The accuracy of the source text and hadith number.
- The clarity of the answer in the Indonesian context.

The assessment results are presented in Table 4.

Table 3. Assessment results.

Assessment Aspect	Average Score (0–100)
Appropriateness of Meaning	95
Accuracy of Source	93
Clarity of Language	96
Average Total	94.6

The KutubBot chatbot, using the combination of $IHS_{\delta}F$ and the Transformer QA model, produces answers that are not only technically accurate but also well-understood by general users. The context of the hadith is maintained, and the system is able to automatically provide biblical source references.

4 Conclusion

This research has successfully developed a Transformer-based Question Answering Chatbot system that automatically, relevantly, and contextually answers hadith-related questions. This system utilizes the Bidirectional Encoder Representations from Transformers (BERT) architecture combined with a new term weighting method, namely Inverse Hadith Space Density Frequency ($IHS_{\delta}F$). The $IHS_{\delta}F$ concept is an extension of Inverse Class Space Density Frequency ($ICS_{\delta}F$) by adding the dimension of hadith space density, so that the weight of each term is not only influenced by its appearance in a particular category, but also by its distribution among various hadith books.

Using a dataset of 63,000 hadiths from nine major books (Kutub al-Tis'ah), the system was successfully trained and tested with excellent results. Test results showed that KutubBot achieved good results, with an 97.8% accuracy, 90.3% average precision, 93.45% recall, and an F1-score of 91.86%. These values represent a significant improvement over the TF-IDF and $ICS_{\delta}F$-based models used for comparison.

Qualitative testing results also show that the system not only provides lexically relevant answers but also understands the semantic context of user questions. For example, the system can distinguish similar meanings across hadiths from different books and still provide answers appropriate to the question's context. Thus, applying a Transformer enriched with $IHS_{\delta}F$ weighting can significantly improve semantic understanding and answer relevance.

Overall, this research demonstrates that a combination of a term-weighted statistical approach and a Transformer-based deep learning approach can produce a more adaptive, accurate, and relevant hadith question-and-answer system within the context of Islamic scholarship. These results are expected to contribute significantly to the development of Natural Language Processing (NLP) technology for religious texts, particularly in the fields of hadith studies and digital Islamic legal sources.

References

1. E. Nasarian, R. Alizadehsani, U. R. Acharya, K. L. Tsui, Designing interpretable ML system to enhance trust in healthcare: A systematic review to proposed responsible clinician-AI-collaboration framework. Inf. Fusion. 108, 102412 (2024). https://doi.org/10.1016/j.inffus.2024.102412
2. R. Deberdt *et al.*, Artificial intelligence and ESG in resources-intensive industries: Reviewing the use of AI in fisheries, mining, plastics, and forestry. Extr. Ind. Soc. 23, 101690 (2025), https://doi.org/10.1016/j.exis.2025.101690
3. A. Casheekar, A. Lahiri, K. Rath, K. S. Prabhakar, K. Srinivasan, A contemporary review on chatbots, AI-powered virtual conversational agents, ChatGPT: Applications, open challenges and future research directions. Comput. Sci. Rev. 52, 100632 (2024), https://doi.org/10.1016/j.cosrev.2024.100632
4. A. Farea, F. Emmert-Streib, Understanding question-answering systems: Evolution, applications, trends, and challenges. Eng. Appl. Artif. Intell. 156, 110997 (2025). https://doi.org/10.1016/j.engappai.2025.110997
5. X. Cheng, X. Zhang, B. Yang, Y. Fu, An investigation on trust in AI-enabled collaboration: Application of AI-Driven chatbot in accommodation-based sharing economy. Electron. Commer. Res. Appl. 54, 101164 (2022). https://doi.org/10.1016/j.elerap.2022.101164
6. L. Pascazio *et al.*, "Question-answering system for combustion kinetics," *Proc. Combust. Inst.*, vol. 40,

no. 1–4, p. 105428, 2024, https://doi.org/10.1016/j.proci.2024.105428

7. K. Du, Y. Zhao, R. Mao, F. Xing, E. Cambria, Natural language processing in finance: A survey. Inf. Fusion. 115, 102755 (2025). https://doi.org/10.1016/j.inffus.2024.102755
8. P. Upadhyay, R. Agarwal, S. Dhiman, A. Sarkar, S. Chaturvedi, A comprehensive survey on answer generation methods using NLP. Nat. Lang. Process. J. 8, 100088 (2024). https://doi.org/10.1016/j.nlp.2024.100088
9. M. S. Salim, S. I. Hossain, T. Jalal, D. K. Bose, and M. J. I. Basher, LLM based QA chatbot builder: A generative AI-based chatbot builder for question answering. SoftwareX. 29, 102029 (2025). https://doi.org/10.1016/j.softx.2024.102029
10. F. Caccavale, C. L. Gargalo, J. Kager, S. Larsen, K. V. Gernaey, U. Krühne, ChatGMP: A case of AI chatbots in chemical engineering education towards the automation of repetitive tasks. Comput. Educ. Artif. Intell. 8 (2025). https://doi.org/10.1016/j.caeai.2024.100354
11. A. Abdi, S. Hasan, M. Arshi, S. M. Shamsuddin, N. Idris, A question answering system in hadith using linguistic knowledge. Comput. Speech Lang. 60, 101023 (2020). https://doi.org/10.1016/j.csl.2019.101023
12. A. B. Kamran, B. Abro, A. Basharat, SemanticHadith: An ontology-driven knowledge graph for the hadith corpus. J. Web Semant. 78, 100797 (2023). https://doi.org/10.1016/j.websem.2023.100797
13. S. S. Sazali, N. A. Rahman, Z. A. Bakar. Characteristics of Malay translated hadith corpus. J. King Saud Univ. - Comput. Inf. Sci. 34, 5, 2151–2160 (2022). https://doi.org/10.1016/j.jksuci.2020.07.011
14. J. J. Kim, H. U. Kim, J. Adamowski, S. Hatami, H. Jeong, Comparative study of term-weighting schemes for environmental big data using machine learning. Environ. Model. Softw. 157, 105536 (2022). https://doi.org/10.1016/j.envsoft.2022.105536
15. T. Dogan, A. K. Uysal, On Term Frequency Factor in Supervised Term Weighting Schemes for Text Classification. Arab. J. Sci. Eng. 44, 11, 9545-9560 (2019). https://doi.org/10.1007/s13369-019-03920-9
16. M. Okkalioglu, TF-IGM revisited: Imbalance text classification with relative imbalance ratio. Expert Syst. Appl. 217, 119578 (2023). https://doi.org/10.1016/j.eswa.2023.119578
17. S. Deo, D. Banik, P. K. Pattnaik, Customized long short-term memory architecture for multi-document summarization with improved text feature set. Data Knowl. Eng. 159, 102440, (2025). https://doi.org/10.1016/j.datak.2025.102440
18. K. Peyton, S. Unnikrishnan, A comparison of chatbot platforms with the state-of-the-art sentence BERT for answering online student FAQs. Results Eng. 7, 100856 (2023). https://doi.org/10.1016/j.rineng.2022.100856
19. M. S. Islam, K. M. Alam, Sentiment analysis of Bangla language using a new comprehensive dataset BangDSA and the novel feature metric skipBangla-BERT. Nat. Lang. Process. J. 7, 100069 (2024). https://doi.org/10.1016/j.nlp.2024.100069
20. C. Aparna, K. Rajchandar, A robust solution for recognizing accurate handwritten text extraction using quantum convolutional neural network and transformer models. Comput. Electr. Eng. 120, 109794. (2024). https://doi.org/10.1016/j.compeleceng.2024.109794
21. J. Yang *et al.*, BERT and hierarchical cross attention-based question answering over bridge inspection knowledge graph. Expert Syst. Appl. 233, 120896 (2023). https://doi.org/10.1016/j.eswa.2023.120896
22. Z. Chen *et al.*, Improving BERT with local context comprehension for multi-turn response selection in retrieval-based dialogue systems. Comput. Speech Lang. 82, 101525 (2023). https://doi.org/10.1016/j.csl.2023.101525
23. Z. Li, K. Xu, Y. Liang, Y. Wang, Z. Zou, DiffuStory : Improving text diffusion models for creative story generation with contrastive learning and decoder-decoder Transformers. Expert Syst. Appl. 130154,(2025). https://doi.org/10.1016/j.eswa.2025.130154
24. I. C. Rico, J. P. Espada, Expert system for extracting keywords in educational texts and textbooks based on transformers models. Expert Syst. Appl. 282, 127735 (2025). https://doi.org/10.1016/j.eswa.2025.127735.
25. H. Pan, B. Teng, Z. Li, Y. Fu, L. Li, A hybrid TH-LSTM-Transformer Model for text generation from EEG signals during imagined character speech. Biomed. Signal Process. Control, 113, 10887 (2026), https://doi.org/10.1016/j.bspc.2025.108871
26. J. Chen, P. K. Kudjo, S. Mensah, S. A. Brown, and G. Akorfu, An automatic software vulnerability classification framework using term frequency-inverse gravity moment and feature selection. J. Syst. Softw. 167,110616, (2020). https://doi.org/10.1016/j.jss.2020.110616
27. J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, C. J. Pérez, The impact of term-weighting schemes and similarity measures on extractive multi-document text summarization. Expert Syst. Appl.169, 114510 (2021). https://doi.org/10.1016/j.eswa.2020.114510
28. Z. Tang, W. Li, Y. Li, An improved supervised term weighing scheme for text representation and classification. Expert Syst. Appl. 189, 24, 115985 (2022). https://doi.org/10.1016/j.eswa.2021.115985
29. J. Choi, S. W. Lee, Improving FastText with inverse document frequency of subwords. Pattern Recognit. Lett. 133, 165–172 (2020).

https://doi.org/10.1016/j.patrec.2020.03.003

30. L. Chen, L. Jiang, C. Li, Using modified term frequency to improve term weighting for text classification. Eng. Appl. Artif. Intell.101, 104215 (2021). https://doi.org/10.1016/j.engappai.2021.104215
31. A. Thakkar and K. Chaudhari, Predicting stock trend using an integrated term frequency–inverse document frequency-based feature weight matrix with neural networks. Appl. Soft Comput. J. 96, 106684 (2020). https://doi.org/10.1016/j.asoc.2020.106684