sciendo INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS

Article | DOI: 10.2478/ijssis-2024-0018 Issue 1 | Vol. 17 (2024)

Medical nearest-word embedding technique implemented using an unsupervised machine learning approach for Bengali language

Kailash Pati Mandal[1,*], Prasenjit Mukherjee[1], Devraj Vishnu[2], Baisakhi Chakraborty[1], Tanupriya Choudhury[3,4,*] and Pradeep Kumar Arya[5,*]

[1]Computer Science and Engineering Department, National Institute of Technology, Durgapur, India

[2]School of Computing Science and Engineering, VIT Bhopal, Bhopal, India

[3]CSE Department, Graphic Era Deemed to be University, Dehradun 248002, Uttarakhand, India

[4]CSE Department, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, Maharashtra, India

[5]School of Computer Science Engineering and Technology (SCSET), Bennett University, Greater Noida, 201310, Uttar Pradesh, India

*E-mails: biltu.cse@gmail.com, tanupriya1986@gmail.com, pradeeparya25@@gmail.com

Received for publication May 13, 2023.

Abstract

The rapid growth of natural language processing (NLP) applications, such as text summarization, speech recognition, information extraction, and machine translation, has led to the development of structured query language (SQL) for extracting information from structured data. However, due to limited resources, converting Natural Language (NL) queries to SQL in Bengali is challenging. This article proposes an unsupervised machine learning model to find semantically Bengali closed words that can generate SQL from NL queries in Bengali. The main objective of the proposed system is to provide support in the creation of patient-oriented explanations and educational resources by simplifying intricate medical terminology. The major findings of the proposed system are as follows: The use of machine translation in the field of medicine facilitates the dissemination of healthcare information to a diverse international audience and improves the performance of entity recognition tasks, including the identification of medical conditions, drugs, or procedures within clinical notes or electronic health data. This system allows a naive user to extract health-related information from a healthcare-structured database without any knowledge of SQL. The system accepts a query and generates a response according to the query in Bengali language. Query tokenization and stop word removal are carried out in the preprocessing stage, and unsupervised machine learning techniques are implemented to process the input query sentence. Tokenized words are converted into vectors using the skip-gram model, with noise-contrastive estimation (NCE) applied to discriminate between actual and irrelevant words. Stochastic gradient descent (SGD) optimizes the model by randomly choosing a small amount of data from the dataset and using cosine similarity to measure closer words. The semantically closer words are found using an unsupervised learning method to generate the SQL.

Keywords

natural language processing, skip-gram model, noise-contrastive estimation, stochastic gradient descent, structured query language

1. Introduction

The database is one of the important data repositories for a computerized system. Information retrieval from the database is an essential task for a computerized system. Structured query language (SQL) is a formal language for accessing the database. Therefore, a

Open Access. Published by Sciendo. CC BY-NC-ND
© 2024 Mandal et al. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License https://creativecommons.org/licenses/by-nc-nd/4.0/

naive user is unable to retrieve information from the database without knowledge of the database query language.

A natural language processing (NLP) system provides freedom for users to extract information from databases using their native language. A naive user can extract information without any technical knowledge. According to the huge growth of machine learning (ML)-based applications in NLP, various NLP-based systems have been developed or are in the development phase. These systems are useful to reduce the gap between human-computer interactions. A ML-based NLP system automates the query conversion process. Unsupervised ML models are very useful for the implementation of NLP-based applications because they work efficiently for unlabeled data. Different machine learning models have been used to develop NLP-based systems. Today, NLP-based systems in the native language are very useful because people can interact with these systems in their own language. Regional language based query–response models have not developed sufficiently. Rural people are not flexible to interact with an interface of the NLP system in English. This is the gap in NLP-based query response models.

Bengali is one of the most widely used vernacular languages in India. Approximately 234 million use the Bengali language as a medium of communication [1]. It was noted that the English language is used in most NLP-based systems. Maximum Structure, grammar, and WordNet databases are developed in the English language, and these kinds of resources are easily available to develop an NLP-based system. Today, vernacular language-based NLP models are widely used in the research community. The performance of different Bengali word embedding techniques such as word2Vec, GloVe, and FastText was calculated based on their syntactic and semantic relatedness. The experiment was performed on 180 million Bengali words. Extrinsic performance provides better results than intrinsic performance, as discussed in a previous study [2]. A Bengali text document classifier was developed using GloVe embedding and a very deep convolution neural network (VDCNN). Embedding parameter identification (EPI) performs well in low-resource languages like Bengali. The classifier showed that the GloVe embedding model performs better than other embedding models on Bengali text, as implemented in a previous study [3]. The detection of abuse words in Bengali text was tested using different machine algorithms such as linear support vector classifier (LinearSVC), logistic regression (Logit), multinomial naive Bayes (MNB), random forest (RF), artificial neural network (ANN), recurrent neural network (RNN) combined with long short-term memory (LSTM). The newly introduced stemming rules give better results in the Bengali language with RNN, as discussed in a previous study [4]. The Bengali language-based question-answering system was implemented using the deep learning method. This system was developed using the SQuAD 2.0 dataset and tested on the Bengali Wikipedia human-annotated dataset as explained in [5]. The sentimental analysis was conducted on Bengali sports news using convolutional neural network (CNN), multilayer perceptron, and LSTM, as explained in a previous study [6]. The news article classification system was implemented using the Multi-Label K-Nearest Neighbors (ML-KNN) algorithm and neural network. A multilabel classification was carried out on Bengali text to improve the performance of various Bengali newspaper portals. A count vectorizer was used for Bengali word embedding to prepare the dataset. The dataset is used for training and testing purposes of the proposed model, as described in a previous study [7]. The Bengali keyword extraction [8] from a document was developed using an unsupervised machine learning technique. The system performance is influenced by word parsing, stemming, and excluding stop words. The Bengali keyword extraction system achieved 87%, as explained in a previous study [8]. Various ML models were used to develop NLP-based systems in Bengali. SQL generation from NL query in Bengali is a good approach in NLP research. The ML models can be broadly used to develop a query–response system that can handle queries in Bengali. Now, it is challenging to develop a query–response system in vernacular languages like Bengali due to insufficient resources. Therefore, we propose a query–response model that can handle queries in Bengali. An unsupervised ML method was used to generate responses from NL queries in Bengali. The proposed work is a query processing system that converts Bengali language queries to SQL. This system enables a user to access the database using a Bengali language query. The naive users extracted information from the database using the Bengali language. The proposed system uses an unsupervised ML technique (skip-gram) for identifying all possible nearest words for each word present in the query. The word and its semantically closer words are used to detect entities and attributes. A set of predefined rules was considered for SQL generation. The generated SQL retrieves desired information from the clinical database. The clinical database consists of a hospital, doctor, and department tables.

The rest of the article is arranged as follows. Details of various NLP systems are discussed in the Related Work section along with a comparative study. The main objective of the proposed research is narrated in The Intended Goal of the Proposed System section. The architecture of the proposed system discusses the proposed system's architecture. The analysis of the proposed system is explained in the Result and Discussion section. The limitation of the proposed system is given in the Limitations section, and future development of this system is stated in the Future Works section. Finally, the proposed work is concluded in the Conclusion section.

2. Related work

The research community has unsolved issues in handling NL queries. Many NLP-based systems have been developed that are able to handle NL queries. Today, ML algorithms are more commonly used in solving critical tasks like parsing, keyword extraction, related word extraction, semantic similarity findings, and the sentiment of word finding in NLP. Different types of ML models are used according to the problem statement in NLP. Various NLP-based systems are discussed in this section that are well-developed, and each system is elaborated with its inside description. Over the past decades, the creation of an NL interface to a database is difficult. The proposed system [9] was implemented to generate responses from complex databases. The neural network was used to implement this system [9]. The user can extract information from the complex database without prior knowledge of SQL. This system reads NL queries and can generate responses without any manual intervention as in [9]. An NLP-based system [10] has been proposed where the NL query is transformed into SQL to extract information from the database efficiently. The database may contain numbers, text, or images, but extraction of numbers, text, or images is associated with the query language. Information extraction from databases like MySQL, PostgreSQL, or MongoDB is not an easy task for naive users. The proposed system [10] uses Elastic search which is able to extract information from the descriptive column of a database. The main aim of this research is to optimize the SQL queries from natural language, as mentioned in a previous study [10]. Another approach is stated here about SQL generation. SQL is used for managing or manipulating data from the relational database management system. However, a user with a technical background and proficiency in SQL may communicate with the database. Designs such as natural language interface to database (NLIDB) were created to enable natural language database querying. This bridging is demonstrated by the "keyword mapping" process, which entails mapping specific keywords from the original NLQ to database objects like relations, attributes, or values of attributes, as described in a previous study [11]. Kaufmann et al. [12] described in-database machine learning (IDBML), a novel method in SQL-based ML. SQL code was generated using Python language with macros in Jinja2 as a process implementation. This process implementation was described by the multidimensional histogram (MDH) probability estimation in SQL. Another implementation was carried out using the equal quantized rank (EQR) variable-width binning method. The accuracy and time were calculated for this proposed model [12], where it has been observed that MDH probability estimation is significantly better than the naive Bayes. This model is efficient in generating SQL code in the field of big data with good accuracy, as mentioned in a previous study [12].

Web interfaces to databases are widely used today for data searching procedures. Most of the interfaces are based on keywords or a combination of keywords. Natural language-based data retrieval systems are important for naive users to interact with the databases without SQL knowledge. Bai et al. [13] have proposed an NLP model based on the IRNet framework. This model is directly related to the natural language query and database. The database entities and attributes have been encoded by the gated graph neural network (GGNN). The database values were used in the prediction model for identifying the table and column to prepare SQL statements to extract information from databases, as mentioned in a previous study [13]. The topic recognition model identifies and categorizes news topics from a large amount of web information. Topic modeling is widely applied in online public opinion monitoring, news recommendations, online examinations, etc. Topic recognition is a challenging task considering accuracy, for which the text representation and topic recognition method has been described Eminağaoğlu et al. [14].

Words can be converted into vectors, which are useful in training ML models in NLP. Word-to-vector representation is a part of NLP where ML models are correlated. According to Tang et al. [15], semantic-rich word vectors are used in a model to get better results through effective word vector training by using the skip-gram algorithm. Each word was distributed according to the context to put them into the semantic space of the word where it was modeled. In the training process, the random gradient descent

method was used to improve the vector representation of words, as mentioned in Tang et al. [15]. The different word embedding techniques were examined for the vector representation of Bengali words. A total 500,000 Bengali articles were collected from different sites on the Internet, among which 105,000 articles were randomly tested. Rafat et al. have claimed that the fastText word2vec model provides good results [16]. The sentimental analysis of Bengali text is a promising area for online business. However, the sentimental analysis of the Bengali digital data does not yield a good result due to the unavailability of a good sentimental analyzer. Sumit et al. [17] have used the word2vec model for representation vectors from Bengali text where skip-gram has been used to find out the close words to understand the sentiment of a Bengali sentence. The skip-gram and word2vec model provide 83.79% accuracy in the sentimental analysis of the Bengali data [17]. The vector space representation of continuous bag of words (CBOW) and skip-gram were enhanced using an average weighted n-gram technique. In a previous study, the optimal neural network and hybridization of the various known methods have been applied to the distributed representation of Kannada words [18]. The word sense disambiguation system in Bengali were designed using Levenshtein distance and cosine similarity, which were used to detect the ambiguous word and actual sense of a word, respectively. The experiment was performed on a corpus consisting of 3860 sentences, which achieved 80.82% accuracy, as discussed in a previous study [19]. Text representation was based on word embedding enhancement, where the probabilistic topic model Latent Dirichlet Allocation (LDA) was used in topic recognition from news text. Word embedding techniques like word-2vec and GloVe were used for extraction, integration, semantic knowledge, and syntactic relationship creation of the text. Well-known ML classifiers were used to recognize the categories of news topics, as described in a previous study [20].

Named-entity recognition is an important work in NLP, and many NLP-based systems use entity recognition models. Supervised learning methods were used to frame Named Entity Recognition (NER) systems where a large dataset is required with label data. Two problems were identified in the supervised dataset that may contain noise as the first problem and the degraded performance of the dataset as the second problem. The use of cross-domain data was the main cause of the degradation of dataset performance. Hu et al. [21] have proposed partial learning and reinforcement learning (PARE) methods in their model to overcome the aforementioned problems. Partial learning uses the new label strategy to handle noise in a dataset, whereas the reinforced learning method is used for mismatching data handling inside the dataset. The trans-dimensional random field language model was modified by introducing the exponential tilting form of reference distribution, noise-contrastive estimation (NCE) of the model parameters, normalization constants, and bidirectional feature extraction using the deep CNN and the bidirectional LSTM. The large corpus for trans-dimensional random field language provides better efficiency using NCE, as implemented in a previous study [22].

NLP techniques can be used in the computer security domain to identify threats using past threat records. Malware is a threat to any computer system and is directly related to computer security. Various malware systems have been developed that are static and dynamic types. Most of the malware models are not good according to malware detection. A natural language-based effective malware detection system [23] has been approached that can be able to detect malware efficiently. The proposed system is an Android-based system, and the skip-gram model, which was used to detect malware. An up-to-date Android malware dataset (AMD) and Comodo Android Benign datasets were used in the skip-gram model to identify the malware and its family. Two scenarios were approached in a previous study [23]: The RF ML algorithm was used for classification as a first scenario after completing the feature extraction using skip-gram. In the second scenario, the test dataset was created, which contains the zero-day malware samples. The proposed malware detection system [23] was compared with other similar types of systems using VirusTotal Application Programming Interface (API) according to the performance [23]. Conversational systems was used for the medical domain. NLP methods were used to develop interfaces for patient conversation. A good interface for conversation in NLP is a chatbot, which is well-known in many organizations today. A chatbot service [24] was developed for the Covenant University Doctor (CUDoctor) telehealth system. This service is based on fuzzy logic and fuzzy inference and uses the symptoms of tropical diseases. Twilio, San Francisco, California, United States API was used to connect this service with the main system. The knowledge base was used to train the fuzzy support vector machine (SVM). The knowledge base contains known facts about diseases and symptoms. The SVM was employed to predict the diseases using the symptom dataset, as mentioned

in a previous study [24]. Sometimes, it is unclear why word2vec has these advantageous characteristics. Even though many academics sought to identify the root of the word2vec's effectiveness, many of their works lacked a rigorous mathematical study of the skip-gram model's formulae. Zhang et al. [25] contributed three primary findings: The learning rules for the input and output vectors are derived after looking at the gradient formulae of the skip-gram model. The word2vec technique shows promising results for word embeddings. Third, the best candidate solution constraints for the input and output vectors of the skip-gram model are given in the training corpus. The foundation of statistics and ML is learning a parametric model of data distribution. After a model was learned, it may be applied to produce new data, assess the probability of already-existing data, or be examined for relevant structure, such as conditional relationships between its features. Maximum-likelihood estimation (MLE), one of several statistical techniques created for this issue, has emerged as the preferred technique. Given data samples, it assesses a model's propensity to produce them and keeps the model with the greatest match. However, MLE is constrained by the need to appropriately normalize the parametric model, which may not be computationally viable. An option recently developed under the name of NCE given data samples, creates noise samples and trains a discriminator to learn the data distribution through contrast. As a binary prediction job, its supervised formulation is straightforward to comprehend and straightforward to use as in [26]. The development of people needs the digital platform more and more each day. However, a more basic issue is also being created by it. Spam is a major problem and the finest litmus test for modern security. Spam wires constantly and unrestrictedly disseminate harmful messages to an enormous number of recipients, presenting a serious security issue. At least one of the complemented systems in ML is prone to security difficulties for spam with a thorough degree of techniques (ML). One aspect of the supervised ML problem is spam disclosure. This study introduces stochastic gradient descent (SGD) frameworks to the spam security challenge, directing malicious comments. For this setup, a targeted analysis of over 40k spam comment datasets is employed. The remaining 10k are used for testing, while the remaining two 30k plans are segregated and used for training as in [27].

A comparison study was conducted with different NLP systems that are related to SQL generation. Table 1 contains the details of each system and is compared with the proposed system.

Table 1: Comparative study of the very same design system with the proposed work.

Author(s)	Methodology applied in near identical system	Methodology applied in proposed system
Arefin et al. [28]	(1) In the article [28], the pre-processing stage utilizes a lowercase conversion, removing escaped words, tokenization, and PoS tagging techniques. (2) The word similarity has been used to select closely related words. (3) The Jaro-Winkler matching algorithm and the Naive method Bayes are utilized to develop this system as in the article [28].	(1) The proposed system employs tokenization, stop-words removal, and stemming in the pre-processing stage. (2) This system uses semantic similarity to select entities and attributes that help to frame the SQL. (3) The skip-gram model is used to detect close related words whereas Noise Contrastive Estimation is used to remove irrelevant words. Stochastic Gradient Descent is employed to optimize the proposed model.
Liu et al. [29]	(1) The proposed model contains three layers that are Encoder layer, parse layer and output layer. (2) The encoder is used to encode the input information where the parse layer is responsible for connecting vectors of various parts. (3) The output part is handling the logit output of this model.	(1) The machine learning model part is divided into three layers namely input layer, hidden layer, output layer. (2) In this case each word is assigned a unique value to encode the information. (3) The cosine similarity captures semantically close words.

(*Continued*)

Table 1: Continued.

Author(s)	Methodology applied in near identical system	Methodology applied in proposed system
Sanyal et al. [30]	(1) This system converts queries from the English language to SQL. (2) The work [30] is related to the high performance of the SQL generation. (3) Tokenization, stop words removal, parsing, lexical analysis, synonym detection and Formation, and filtered word mechanism of NLP to implement the work [30].	(1) The proposed system converts the Bengali queries to SQL. (2) The unsupervised machine learning model has been used to generate SQL from NL queries. (3) NLP based data preprocessing task and the Skip-gram model have been applied to implement the proposed system.
Sugandhika and Ahangama. [31]	(1) The XML file contains the metadata of a particular database where the XML extractor works to read this XML file for metadata. This metadata has been used to generate SQL because that contains information about table names, column names, operator details, etc. (2) The BASIC clause generator is used to extract elements that are useful to form the BASIC clause of the SQL. (3) The column names, column value, relational operators, and concatenating operators are used to frame SQL.	(1) The unlabelled Bengali text has been used to train the skip-gram model to find out the related words. (2) A set of predefined rules has been applied to form SQL with 'SELECT', 'WHERE', IN, AND, OR, etc. clauses. (3) Semantically close words have been used to predict the entity and attribute of a particular table.
Pal et al. [32]	(1) The proposed model [32] is a deep learning-based model that handles natural language questions and generates valid SQL. (2) According to the sensitive nature of some databases, the data privacy approach has been included in this proposed model [32]. (3) The model [32] used vectors that are RoBERTa embedding and data-agnostic knowledge vectors. Vectors are passed through some sub-models that are also LSTM-based models to predict the final SQL query.	(1) The proposed system uses the unsupervised skip-gram model to covert Bengali NL queries to SQL. (2) Noise Contrastive Estimation (NCE) is used to discriminate between actual data and the noise of the data. (3) A unique value is assigned to each and every unique word to form a vector. Vectors have been used as input in the skip-gram model to identify the close words.
Huo et al. [33]	(1) In SyntaxSQLNet, a Bi-directional LSTM is used to encode a natural language phrase as in [33]. (2) This approach incorporates both the global table information and the local column information that is used as input to the BiLSTM. (3) SQL-specific tree-based decoder has been used in the proposed model to understand the SQL structure. The COL module is used to predict the column and this model has been improved here.	(1) The proposed system encodes Bengali phrases with the help of word embedding method. (2) The created dictionary is employed to train the skip-gram model. (3) Output of the skip-gram model is used to identify the entity and attributes in a predefined healthcare database.

3. The intended goal of the proposed system

The intended goal of the proposed system is to generalize information access from databases using vernacular language. Natural language queries in English to SQL conversion is a common research area to reduce the gap between humans and computers, but any vernacular language to SQL conversion is a difficult task. Vernacular languages are not the same according to their grammar formation. Steps of NLP like sentence extraction, keyword extraction, tokenization, lemmatization, named-entity recognition, semantics are difficult in any vernacular language.

Additionally, vernacular language components are quite limited; therefore, establishing any NLP step is technically difficult. Most of the NLP systems that are related to natural language queries to SQL generation are dependent on English. However, users from rural areas are not flexible in English. They are flexible in their native language. Access to healthcare information using vernacular language is an interesting research topic, and it will be helpful for rural people. The intended goal of this research is to develop a query response model that could receive NL queries in Bengali and generates responses in Bengali. This system has been prepared in the healthcare domain. The healthcare database contains information in Bengali. The responses have been generated after SQL conversion from an NL query in Bengali. The proposed system will be helpful for rural people who are using Bengali as their communication language in the healthcare domain.

4. Architecture of the proposed system

The article proposes an unsupervised ML health-related query processing system (HRQPS) to handle Bengali natural language queries with semantically close words. The flowchart of the proposed system is given in Figure 1.

The user posted a health-related query in the Bengali language in HRQPS. The query was sliced into the token(s). Stop word(s), and nominal inflections, was(were) removed from the query token. Each and every token was passed through the skip-gram model for the detection of semantically related words. This ML model was trained with a health-related dataset so that it can identify all semantically related word(s) for a token. The cosine distance similarity has been applied to measure the semantically closer words. These semantically closer words help identify entities and attributes. Entity and attributes were used to form SQL. Finally, the proposed system executes the SQL and retrieves desired results from the clinical database. The working principle of the proposed system has been divided into three parts: the proposed algorithm, execution steps of the skip-gram model, and the structure of the clinical database.

A. Proposed algorithm

Figure 2 shows the workflow diagram for the proposed system. The algorithm of the proposed system

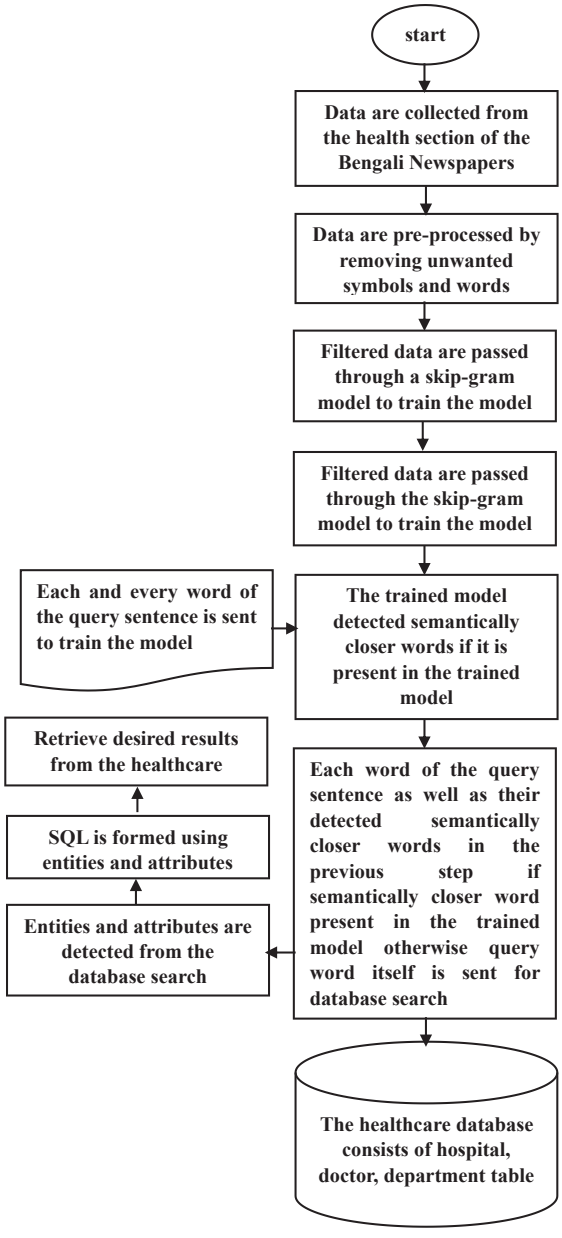

Figure 1: Flowchart of the proposed system.

is given as follows. The website https://ipa.bangla.gov.bd/ was used to generate the IPA notation for Bengali terms.

Step 1:
In Bengali, a user makes a health-related question to the proposed system. The query was given as follows:
Query string = "হাওড়াতে কিডনির পাথরের চিকিৎসার জন্য কি কি হাসপাতাল আছে?" ("haʊɽaṭe kidnir paṯʰorer cikiʦar ɟonno ki haʃpaṯal acʰe?") (Which

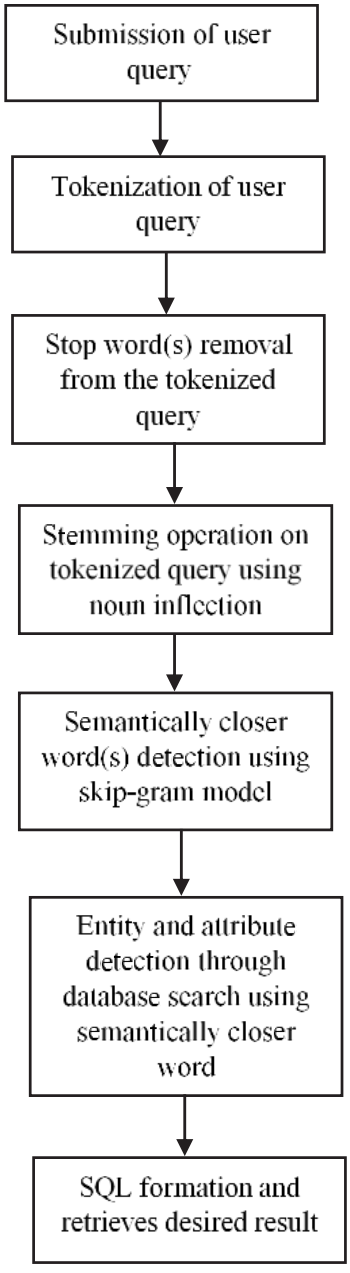

Figure 2: Workflow diagram of the proposed system.

hospitals are there for the treatment of kidney stones in Howrah?).

Step 2:

The Bengali query was divided into linguistic undivided smallest parts by considering the white space present in the sentence. These token(s) are stored in a list.

Token list = ["হাওড়াতে" (Howrah), "কিডনির" (kidney), "পাথরের" (stones), "চিকিৎসার" (treatment), "জন্য" (for), "কি" (which), "কি" (which), "হাসপাতাল" (hospital), "আছে" (are there)] (["ha͡o̯ɽat̪e", "kidnir", "pat̪ʰorer", "cikit̪ʃar", "ɟonno", "ki", "ki", "haʃpat̪al", "acʰe"]).

Step 3:

In the domain of text processing and NLP, the term "stop words" pertains to the frequently occurring words within a given language. In the preprocessing step, it was common practice to exclude these occurrences due to their high frequency, which results in a smaller amount of usable information for analysis. This means that the system we are developing has a list of commonly used Bengali words that it will ignore during text processing. This was likely because these words do not contain significant meaning on their own and could potentially skew the results if included. For instance, if we are building a system to analyze Bengali text and identify semantically close words, including stop words in our analysis could misleading results. A word like "এবং" (and), "অথবা" (or) might appear very frequently simply because it's a common conjunction. The proposed system contains a predefined list of Bengali stop words. Each and every token was checked against the predefined list of Bengali stop words. The proposed system removes a token from the token list if it matches the predefined list of Bengali stop words. A total of 398 Bengali stop words were considered in this study. The list of stop words and the list of tokens after filtration are given as follows:

Stop words list = ["আমি", "আমরা", "দ্বারা", "কি", "জন্য", "এবং",...] (["ami", "amra", "d̪ara", "ki", "ɟonno", "eboŋ",...]) ("I", "we", "by", "what", "for", "and"…).

Filtered token list =["হাওড়াতে" (Howrah), "কিডনির" (kidney), "পাথরের" (stones), "চিকিৎসার" (treatment), "হাসপাতাল" (hospital), "আছে"] (["ha͡o̯ɽat̪e", "kidnir", "pat̪ʰorer", "cikit̪ʃar", "haʃpat̪al", "acʰe"]).

Step 4:

The Bengali nominal root word was extracted from the token by removing the nominal inflectional part. The proposed system contains a predefined nominal inflection list of Bengali words. Every filtered token was compared with the predefined nominal inflection list of Bengali words. No pre-developed stemmers were used in this study. The proposed system removes the trailing part from the filtered token if it matches the predefined nominal inflection list of Bengali words. The nominal inflection list and root words are given as follows:

Nominal inflectional = ["ে", "র", "দের", "গুলি", "রে", "তে", …] (["e", "r", "d̪er", "guli", "er", "t̪e",…).

Root word = [“হাওড়া” (Howrah), “কিডনি” (kidney), “পাথর” (stones), “চিকিৎসা” (treatment), “হাসপাতাল” (hospital), “আছ” (are there)] ([“haʊɽa”, “kidoni”, “pat̪ʰor”, “cikit̪ʃa”, “haʃpat̪al”, “acʰo”]).

Step 5:

The extracted six root words are passed through the skip-gram model that helps to detect semantically closer words as discussed in the execution steps of the skip-gram model. The proposed system considers only 15 semantically closer words for each root word that helps to predict entity and attribute.

The skip-gram model does not find any semantically closer word for token “হাওড়া” (Howrah) (“haʊɽa”) because it was a proper noun, as well as the dataset, was small. Therefore, the token “হাওড়া” (“haʊɽa”) itself was sent for database searching.

The skip-gram model detects semantically closer words for the token “কিডনি” (kidney) (“kidoni”). The proposed system selects 15 semantically closer words as shown in Figure 3. The token “কিডনি” (kidney) (“kidoni”) and 15 selected semantically closer words were sent for data searching.

Similar way the proposed system detects semantically closer words for “পাথর” (stone) (“pat̪ʰor”), “চিকিৎসা” (treatment) (“cikit̪ʃa”), “হাসপাতাল” (hospital) (“haʃpat̪al”) as shown in Figures 4–6. These tokens and semantically closer were sent for database searching.

The proposed system does not find any semantically closer words for the token “আছ” (are there) (“acʰo”). Therefore, the token “আছ” (are there) (“acʰo”) itself was sent for database searching.

Step 6:

Every token and its semantically closer words are sent for database searching to predict entities and attributes. If the token or its semantically closer words match with the database value then the corresponding attribute name, as well as an entity, was retrieved.

“হাওড়া” (Howrah) (“haʊɽa”), “কিডনি” (kidney) (“kidoni”), “পাথর” (stone) (“pat̪ʰor”), “চিকিৎসা” (treatment) (“cikit̪ʃa”), “হাসপাতাল” (hospital) (“haʃpat̪al”), “আছ” (are there) (“acʰo”) tokens are sent for database searching for the user submitted query.

The proposed system does not find any semantically closer word for “হাওড়া” (Howrah) (“haʊɽa”); therefore, the token “হাওড়া” (Howrah) (“haʊɽa”) itself was sent for database searching. The token “হাওড়া” (Howrah) (“haʊɽa”) fits with the value of the district_hos attribute of the table hospital. The distrct_hos attribute well as the table name hospital was retrieved.

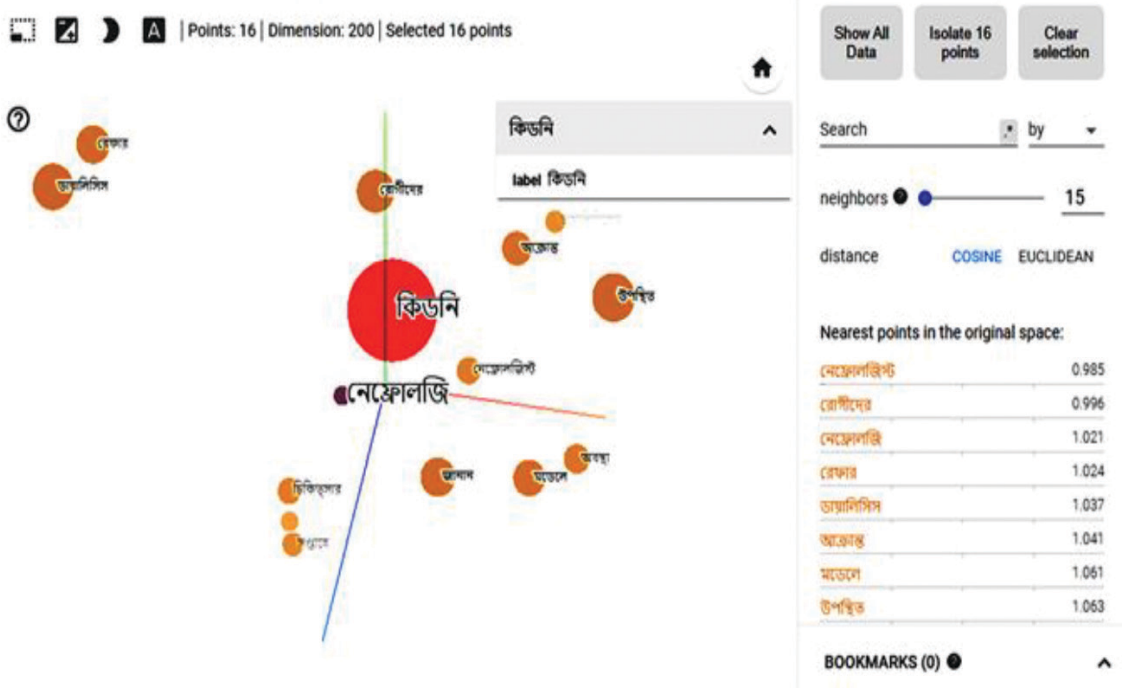

Figure 3: Semantically closer words of কিডনি (kidney).

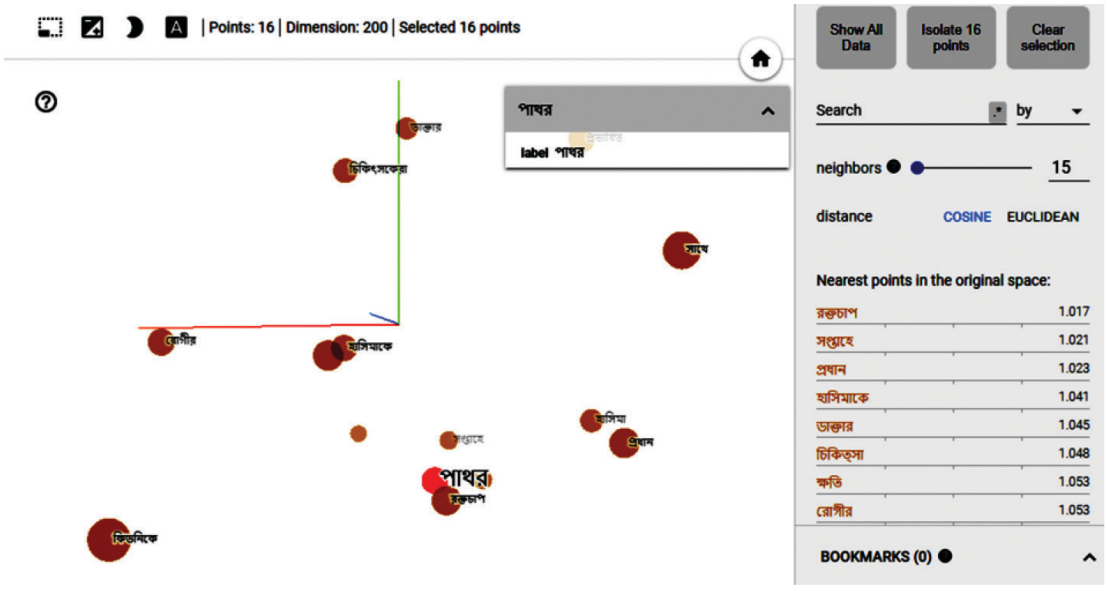

Figure 4: Semantically closer words of পাথর (stone).

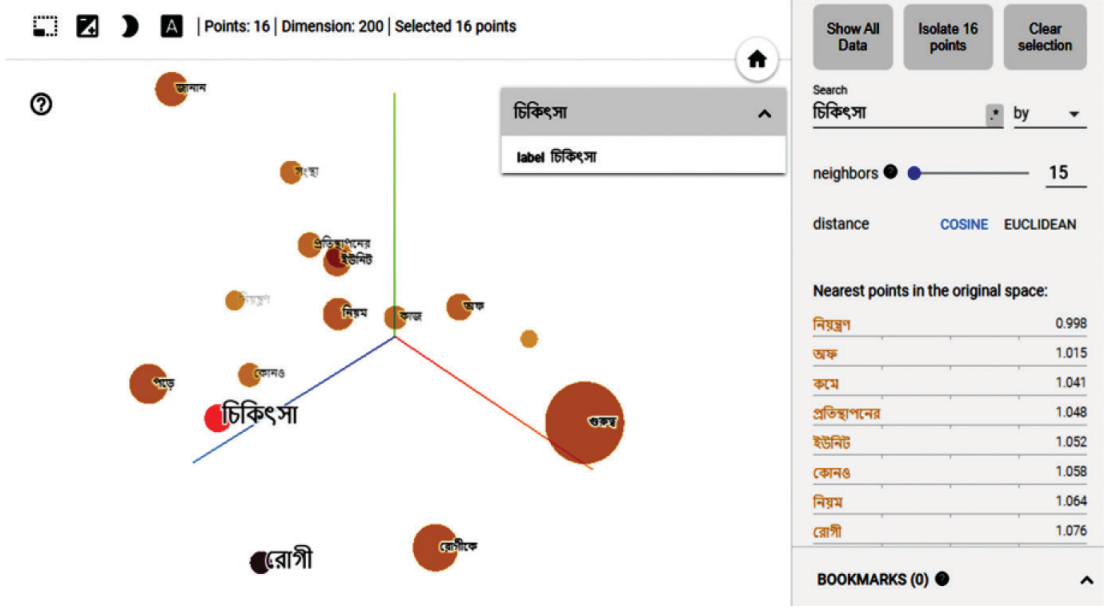

Figure 5: Semantically closer words of চিকিৎসা (treatment).

Similarly, the proposed system performs database searching for remaining all other tokens present in the query.

Step 7:

The entity and attribute are used to form SQL. The generated SQL retrieves the desired result from the clinical database. The query formation structure for the proposed system is given as follows:

SELECT * FROM table1, table2,…tablen, WHERE table1.attribute1=value1 AND table2.attribute2=value2 AND… tablen.attributen=valuen. The proposed system has identified district_hos and name_dept attributes, as well as corresponding tables name hospital, and department, respectively, for user-submitted queries. The table name and their attribute were concatenated by the dot operator and treated as a condition. All conditions were joined with AND operator if there was more than one condition present. The joining operation was performed

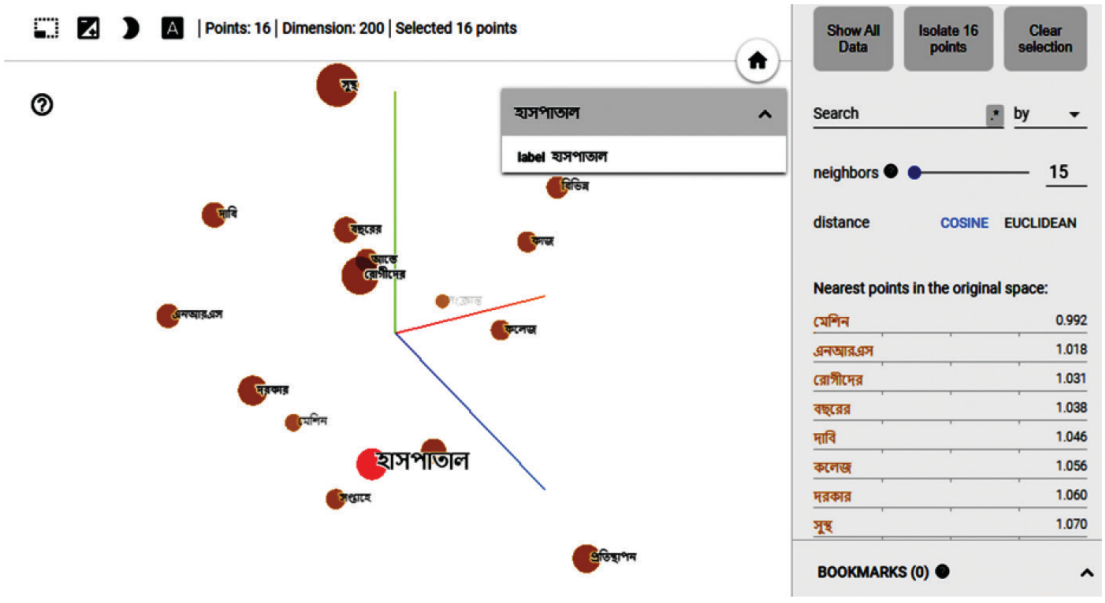

Figure 6: Semantically closer words of হাসপাতাল (hospital).

Conversion of Natural Language Query to SQL:

SELECT * FROM hospital, department, doctor WHERE department.name_dept = 'নেফ্রোলজি' AND doctor.specialist_doc = 'নেফ্রোলজিষ্ট' AND hospital.id_hos = department.id_hos AND hospital.id_hos = doctor.id_hos AND department.id_dept = doctor.id_dept

The Generated Response:

id_hos	name_hos	add_hos	district_hos	state_hos	id_dept	name_dept	id_hos	id_doc	name_doc	qualification_doc	specialist_doc	id_hos	id_dept
8	হাওড়া জেলা হাসপাতাল	আমতা	হাওড়া	পশ্চিমবঙ্গ	80	নেফ্রোলজি	8	1070	সুজন দাশগুপ্ত	এম.বি.বি.এস.	নেফ্রোলজিষ্ট	8	80

Figure 7: Response of user submitted query.

between the department, hospital, and doctor table because the referential integrity existed among the three tables as described in the structure of the clinical database. The join operation was placed after condition(s) with AND operator. The SQL is provided as follows for user-submitted queries:
SELECT * FROM hospital, department, doctor WHERE department.name_dept = "নেফ্রোলজি" (nephrology) AND specialist_doc = "নেফ্রোলজিষ্ট" (nephrologist) AND hospital.id_hos = department.id_hos AND hospital.id_hos = doctor.id_hos AND department.id_dept = doctor.id_dept. The system-generated response is given in Figure 7.

B. Execution steps of the skip-gram model

Skip-gram was the most widely used unsupervised word2vec model. This model determines the closely related words for a given word. The skip-gram can ascertain the most closely related words of any word on its own from unlabeled text [34]. The architecture of the skip-gram model as in Figure 8, as well as the algorithmic steps of the skip-gram model, is given as follows:

1) Data collection
Unstructured data were used to train the skip-gram system. The data of the corpus were

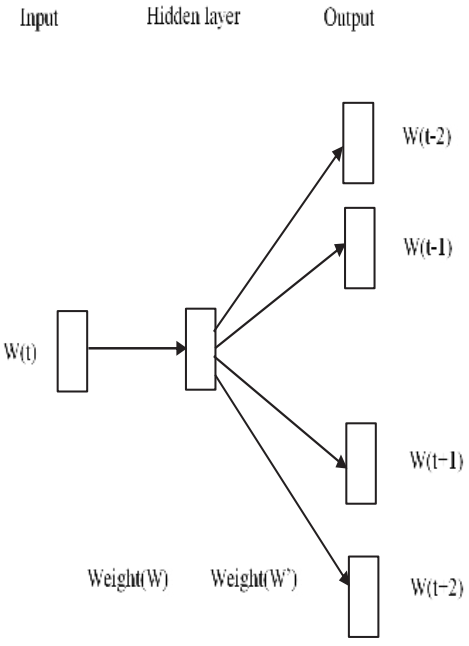

Figure 8: The skip-gram model's architecture.

collected from different Bengali e-newspaper sources. The health section of the e-news was chosen for data acquisition. These newspapers used were Anandabazar patrika, Zeenews, Sangbadpratidin, Bartamanpatrika, Aajkaal, and Bangladesh-pratidin. The collected data were stored in a text file.

2) Data preprocessing
The non-Bengali characters, unwanted symbols, or punctuation marks were eliminated from the text file. The filtered text data were tokenized into word(s) by considering a white space separator. The tokenized words were stored in a list. The proposed system also maintains a predefined list of Bengali stop words. If the tokenized word matches with the stop word list, then that word is removed from the list. The normal distribution was used for word embedding. The normal distribution, sometimes called the Gaussian distribution, is a probability distribution that is symmetric around the mean, demonstrating that data close to the mean occur more frequently than data distant from the mean. The random.normal() function of the TensorFlow [35] was used for word embedding.

3) Input vector for the skip-gram model
A total of 8753 most frequently occurring words were selected from the dataset to prepare the vocabulary. The window size was chosen as three for the skip-gram model. A particular word was fed as input to the skip-gram model as a 1 × 8753-dimension vector with one hot encoding format.

4) Weight matrix of the hidden layer
The embedding size determines the number of neurons in the hidden layer. The used embedding size was 200 means, and 200 neurons were present in the hidden layer. When the input vector comes in the hidden layer, the dot product was performed between the input vector and hidden layer weight matrix. The weight matrix of the hidden layer was initialized to small random values for neural network training. The output of the hidden layer was a dot product between the input vector and weight matrix of the hidden layer i.e. 1 × 200 word vector. The vocabulary size as well as the embedding size determine the dimensions of a weight matrix i.e. 8753 × 200. The weight matrix for the hidden layer is represented by w.

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1200} \\ w_{21} & w_{22} & \cdots & w_{2200} \\ \vdots & \vdots & & \vdots \\ w_{87531} & w_{87532} & & w_{8753200} \end{bmatrix} \quad (1)$$

5) Weight matrix of the output layer
The output of the hidden layer was fed as input to output layer. The output layer vector was calculated by performing dot product between output of hidden layer and weight matrix of output layer. The dimensions of weight matrix were 200 × 8753. The weight matrix for output layer is represented by w'.

$$w' = \begin{bmatrix} w'_{11} & w'_{12} & \cdots & w'_{18753} \\ w'_{21} & w'_{22} & \cdots & w'_{28753} \\ \vdots & \vdots & & \vdots \\ w'_{2001} & w'_{2002} & & w'_{2008753} \end{bmatrix} \quad (2)$$

6) Context word prediction
The softmax function directly calculates the probability of each output vector for a given input [34]. The probability calculation of the softmax function is given as follows:

$$p(Z_i) = \frac{e^{Z_i}}{\sum_{j=1}^{V} e^{Z_j}} \quad (3)$$

In Eq. (3), e^{Z_i} denotes the standard exponential function for the input vector. v denotes vocabulary size.

The softmax activation function was used to predict the multinomial distribution. It was used in a model that deals with multiclass classification problem. The denominator of softmax function is a sum of exponential value of all words in the vocabulary, which is computationally expensive. As the NCE is an unnormalized estimation technique, it estimates the probability of output vector by skipping denominator [36] as well as scaling down the multiclass classification problem to linear regression (binary classification) problem. The NCE calculates loss and adjusts the weight matrix w and w'. The NCE estimation equation is given as follows:

$$logit = \log\left(\frac{P}{Q}\right) = \log(P) - \log(Q) \quad (4)$$

In Eq. (4), P is the actual target word that emerged from the noise distribution and Q denotes noise distribution.

7) Optimization

The SGD optimizer was used to minimize the loss. SGD optimizer gives better result by updating its parameters in each iteration [37]. The equation is given as follows:

$$\theta = \theta - \eta \cdot \Delta_\theta J\left(\theta; x^{(i)}; y^{(i)}\right) \quad (5)$$

In Eq. (5), θ denotes weight matrix. The η represents learning rate. The $\Delta_\theta J$ expresses the gradient of weight matrix. The $x^{(i)}$ denotes training example and $y^{(i)}$ represent label.

C. Structure of the clinical database

The clinical database is a data repository of medical related information. It contains information about hospitals, doctors and departments for a particular area. The hospital table stores information about hospital's name, address, etc. The department table stores details about department's name and allied hospitals, etc. The doctor table contains doctor's name, qualification, etc. The information of clinical database was kept in the format of Bengali language. The proposed system retrieves desired result from the clinical database for health-related query in Bengali language. The concept of clinical database design of the proposed was taken from the research article [38]. These tables were hospital, department, and doctor.

1) Hospital table

The hospital table comprises of hospital's number (id_hos), hospital's name (name_hos), hospital's address (add_hos), hospital's district (district_hos), hospital's state (state_hos) attributes. The "id_hos" attribute is the primary key of the hospital table. Table 2 includes a specific example of the hospital table.

2) Department table

The department table is made up of attributes department's number (id_dept), department's name (name_dept), and hospital's number (id_hos). The "id_dept" and id_hos attributes were the primary key, and foreign key of the department table, respectively. Table 3 includes a specific example of the department table.

3) Doctor table

The attributes of the doctor table were the doctor's number (id_doc), doctor's name (name_doc), doctor's qualification (qualification_doc), doctor's specialization (specialist_doc), doctor's fee (fee), "id_hos" and "id_dept."

Table 2: Structure of hospital table.

id_hos	name_hos	add_hos	district_hos	State_hos
1	মুর্শিদাবাদ জেলা হাসপাতাল (murʃiɖabaɖ ɟela haʃpaṭal) (Murshidabad District Hospital)	লালগোলা (lalgola) (Lalgola) (proper noun)	মুর্শিদাবাদ (murʃiɖabaɖ) (Murshidabad) (proper noun)	পশ্চিমবঙ্গ (poʃcimbɔŋgo) (West Bengal) (proper noun)
8	হাওড়া জেলা হাসপাতাল (hao̯ṛa ɟela haʃpaṭal) (Howrah District Hospital)	আমতা (amoṭa) (Amta) (proper noun)	হাওড়া (hao̯ṛa) (Howrah) (proper noun)	পশ্চিমবঙ্গ (poʃcimbɔŋgo) (West Bengal) (proper noun)

The "id_doc" attribute was the primary key, and "id_hos" and "id_dept" were foreign keys of the doctor table. Table 4 includes an example of the doctor table.

5. Results and discussion

The proposed system accepts NL queries in Bengali and produces responses after processing the queries. The healthcare database was prepared using values in Bengali. Simultaneously, the proposed system was tested using many NL queries in Bengali. The responses were generated successfully according to the NL queries in Bengali. Computer science often involves algorithm-solving problems. The time complexity of an algorithm is linearly dependent on inputs and system configuration. It determines execution duration. Understanding this helps developers choose the most efficient algorithm, often choosing a highly efficient method over a slower one, despite its superior performance. For this system, the time complexity was determined in order to assess performance over time. The F1 score was calculated using precision and recall to understand the accuracy of the proposed system. The algorithmic steps were described as follows:

Step 1: Sign in to the system, and enter the query in Bengali.
After logging in to the system, the user places the query into the system in Bengali language.
Step 2: Tokenization.
The user-submitted query is tokenized into a meaningful linguistic unit called a token.

Step 3: Identification of stop words.
After tokenization each and every token is checked against a predefined Bengali stop word list. If the token matches with the stop word list, then the matched token is removed from the token list.
Step 4: Extraction of root word after removal of the nominal inflectional part.
The filtered token list was compared with a predefined list inflectional list. If the token trailing part of the token matches with inflectional then the trailing part is removed from the token and the root word is extracted.
Step 5: Semantically closer word detection using the skip-gram model.
The root words were passed through a skip-gram model that identifies the semantically closer word.
Step 6: Database searching to identify entities and attributes.
The root word itself and the corresponding detected semantically closer word were sent for database search that predict the entity and attributes.
Step 7: SQL generation and execution.
Entity and attribute form SQL. Finally, the executed by the proposed system.

Time complexity of the proposed system:
The time complexity was estimated for the proposed system.

Step 1: Sign in to the system, then enter the query in Bengali.
Let n number of words be present in the user submitted query. The time complexity was calculated on the proposed algorithm using the above example query. The time was calculated in each algorithmic step that is given as follows:
Step 2: Tokenization.
Let there be n number of the token(s).
The time required to tokenize n tokens = *n* unit time.

Table 3: Structure of department table.

id_dept	name_dept	id_hos
70	নবজাতক (neonate)	7
170	মনোরোগ (psychiatry)	12

Table 4: Structure of doctor table.

id_doc	name_doc	qualification_doc	specialist_doc	id_hos	id_dept
1070	সুজন দাশগুপ্ত (ʃuɟon d̪aʃgupt̪o) (Sujon Dasgupta)	এম.বি.বি.এস. (emo. bi.bi.es.) (M.B.B.S.)	নেফ্রোলজিষ্ট (nepʰrolɟiʃto) (nephrologist)	8	80
1080	সোমেন দে (ʃomen d̪e) (Somen De)	ডি.এম. (di.em.) (D.M.)	নিউরোলজিস্ট (niu̯rolɟist̪) (neurologist)	9	90

Step 3: Identification of stop words.
Let there be m number of stop words in the list of Bengali stop words.
Time is taken for stop word detection = m_{C_n} unit of time = x unit time.
Therefore, the number of token(s) other than stop words = $(n - x)$
Step 4: Extraction of root word after removal of the nominal inflectional part.
Let there be p number of nominal inflectional parts in the nominal inflectional list.
Let q number of nominal inflectional be part in (n-x) number of tokens.
∴ Number of root words = $(n - x) - q$
∴ Time taken for execution of root and removal of the nominal inflectional part.
= $p + q + \{(n - x) - q\}$ unit time.
Step 5: Semantically closer word detection using the skip-gram model.
The time complexity of the skip-gram model is $O(v)$ where v is the size of the vocabulary.
Let there be r number of semantically closer words that are detected by the skip-gram model. Since the proposed system considers only 15 semantically close words for each root.
∴ Time taken = $15r$ unit.
Step 6: Database searching to identify entities and attributes.
Let there be $s1$ and number of rows a $t1$ number of columns in the hospital table.
∴ Time taken for searching $s1t1_{C_{15r}} + s1t1_{C_{(n-x-q)-15r}}$ unit of time.
= $(a + b)$ unit of time (say).
Let there be $s2$ number of rows and $t2$ number of columns in the department table.
∴ Time taken for searching = $s2t2_{C_{15r}} + s2t2_{C_{(n-x-q)-15r}}$
= $(c + d)$ unit of time (say).
Let there be $s3$ number of rows and $t3$ number of columns in the doctor table.
∴ Time taken for database searching = $s3t3_{C_{15r}}$ $s3t3_{C_{(n-x-q)-15r}}$ unit time.
= $(e + f)$ unit time (say).
Step 7: SQL generation and execution.
After the identification of entities and attributes the proposed system generates SQL. The time taken for SQL generation and execution are g and h units of time, respectively.
∴ Time complexity

$$= f(n,m,x,p,q,v,r,s1,t1,s2,t2,s3,t3,a,b,c,d,e,f,g,h)$$

$$= n + x + p + q + \{(n - x) - q\} + 15r + (a + b) + (c + d) + (e + f) + (g + h) + O(v)$$

$$f(n) \cong n + n + n + n + \{(n - n) - n\} + (n + n) + (n + n) + (n + n) + (n + n) + O(v)$$

= 26n + O(n)
= $26n + O(n)$
= $O(n)$
$T(n) = O(n)$, where time unit may be in nano second, μ second or mili second.

Few examples are given to understand the time complexity of the proposed system. A graph is given at the end of this section where time taken line can be seen.

According to the time complexity $O(n)$, n refers to the number of tokens in query. If a query contains 5 number of tokens, then time complexity can be calculated as $O(n) = 5$ where number of tokens (n) is 5.

Q1 = কোথায় হার্টের চিকিৎসা হয়?
Q2 = নবজাতক বিভাগের ডাক্তারগুলির নাম কি?
Q3 = বীরভূম জেলা হাসপাতালে স্ত্রীরোগবিদ্যা বিভাগ আছে কি?
Q4 = হাওড়াতে কিডনির পাথরের চিকিৎসার জন্য কি হাসপাতাল আছে?
Q5 = বাকুড়া, পুরুলিয়া ও নদিয়ার হাসপাতালের মধ্যে অস্থি চিকিৎসা বিভাগ কোথায় আছে?

$O(n)$ represents the time complexity of a function that increases linearly and in direct proportion to the number of inputs. A graph is given in Figure 9 to understand the linearity of the time complexity function. The graph was prepared using Table 5.

Accuracy Calculation of Proposed System:

The confusion matrix is a tabular representation that provides a concise summary of the predictive accuracy of a ML model when evaluated against a specific test dataset. The measurement of classification model performance is frequently employed to assess the accuracy of predicting categorical labels for individual input instances. The matrix presents the quantities of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) generated by the model when applied to the test data. In the context of binary classification, the matrix will have dimensions of 2 ´ 2. The accuracy was calculated using precision and recall where 156 natural language queries in Bengali were used.

The user has supplied 156 different numbers of inquiries into the proposed system. On the basis of their outcomes, four distinct sorts of queries were provided as examples. The result is divided into "Expected output" and "Select output." "Expected output" and "Select output" were also divided

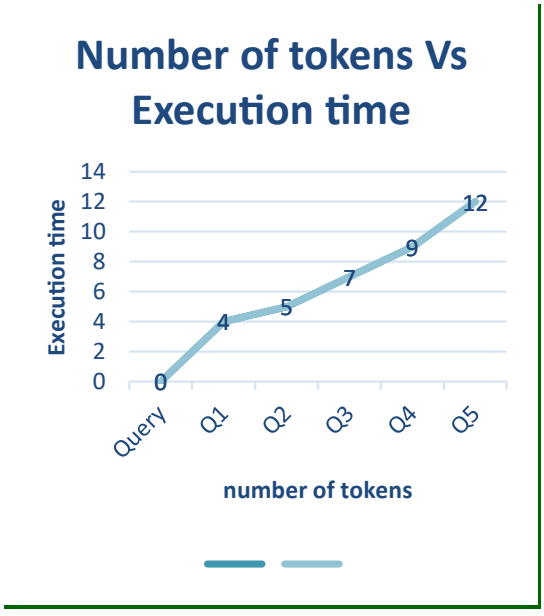

Figure 9: Execution time with respect to number of tokens.

Table 5: Execution time with respect to number of tokens.

Query	Number of tokens	Execution time units
Q1	4	4
Q2	5	5
Q3	7	7
Q4	9	9
Q5	12	12

Table 6: Confusion matrix.

Expected output Vs. Select output	Select output-Positive	Select output-Negative
Expected output-Positive	True positives (TP)-65	False positives (FP)-17
Expected output-Negative	False negatives (FN)-25	True negatives (TN)-49

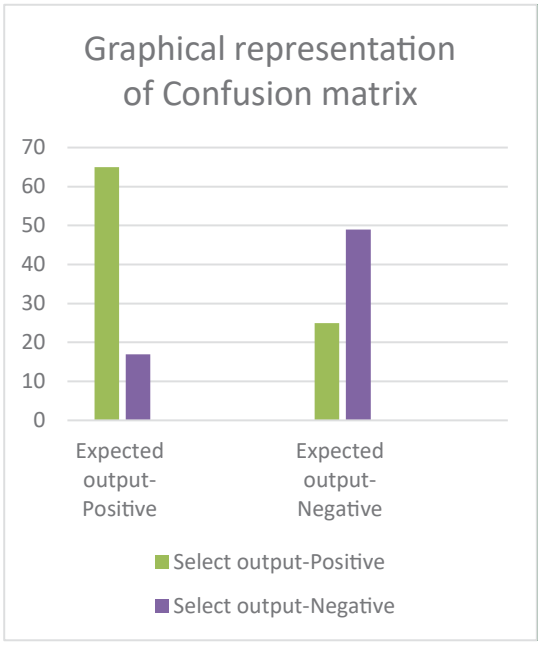

Figure 10: Graphical representation of the confusion matrix.

into positive and negative. Table 6 displays the Confusion Matrix for the proposed system and Figure 10 shows the graphical representation of the confusion matrix.

The following are some examples of classified sample queries:

TP- হাওড়াতে কিডনির পাথরের চিকিৎসার জন্য কি হাসপাতাল আছে? (What hospitals are available in Howrah for the treatment of kidney stone?)

TN- ক্রনিক রোগ আয়ুর্বেদ চিকিৎসা দ্বারা পুরোপুরি সেরে যায় কি? (Is chronic disease fully curable by the treatment of Ayurveda?)

FP- আয়ুর্বেদের হাসপাতাল কোথায় আছে? (Where is the hospital for Ayurveda treatment?)

FN- বীরভূম জেলা হাসপাতালে কি বিভাগ আছে? (What are the departments in the Birbhum district hospital?)

Precision = TP/(TP + FP) = 65/(65 + 17) = 65/82 = 0.79

Our system has a precision of 0.79, which means that 79% of the time when it predicts that a query will be fetched, it is right.

Recall = TP/(TP + FN) = 65/(65 + 25) = 65/90 = 0.72

Our system accurately recognizes 72% of all fetched requests, or recall, of 0.72.

Accuracy = (TP + TN)/(TP + FP + FN + TN) = (65 + 49)/(65 + 17 + 25 + 49) = 114/156 = 0.73.

Our system predicts predictions for fetched queries with an accuracy of 0.73, or 73% accurately.

F1 Score = 2(Recall × Precision) / (Recall + Precision) = (2 × 0.72 × 0.79)/(0.72 + 0.79) = 1.1376/1.51 = 0.75.

Our system's F1 score is 0.75, or 75%, which indicates that it is 75% accurate overall. The performance statistics of the proposed system is given in Table 7.

6. Limitations

The proposed NLP-based system is able to handle NL queries in Bengali. This system also produces responses after converting the NL query to SQL. The SQL generation from NL query is automated and it is the backbone to generate response according to the query. However, the proposed system has few limitations. This system is able to handle only medical related general queries where it does not handle the quantitative query like "How many hospitals are in Bardhaman district for treatment of kidney disease?" and qualitative query like "Which is the best hospital in Birbhum district? Sometimes, the proposed system does not find any semantically closer word for proper noun due to small dataset. The dataset should be modified according to the nature of the queries. The preparation of this kind of dataset to handle qualitative and quantitative queries is another task to implement in future. This system is unable to extract information using NL query in Bengali from the unstructured database.

7. Future works

The proposed system is able to handle general queries but fails to handle qualitative and quantitative queries as mentioned in the limitation of the proposed system. The proposed system can be enhanced to handle quantitative as well as qualitative query in future. The training dataset may be increased so that it can identify proper noun properly. The dataset has to be implemented according to the nature of the queries. The proposed system can be implemented further to extract information from unstructured data and handle multiple queries from multiple domains. Above implementation proposal of this proposed system will be Future work.

Table 7: Performance statistics of the proposed system.

Precision	Recall	Accuracy	F1 score
79%	72%	73%	75%

8. Conclusion

The proposed system is a query–response system. This system transforms Bengali natural language query to its equivalent SQL query. The proposed system incorporates the skip-gram word embedding model that captures the context word(s) for every word present in the query. A small Bengali dataset was prepared to train the skip-gram model. The cosine similarity selects semantically closer context word(s). The proposed system chooses 15 semantically closer context words for database searching that helps to find entity and attribute. The entity and attribute form SQL query and retrieves the desired result.

Acknowledgment

The authors extend deepest regards and heartfelt gratitude to the Department of Computer Science and Engineering, NIT Durgapur, India for providing us with their valuable support and necessary help in the publication of this research paper.

References

[1] O. Sen, et al., "Bangla Natural Language Processing: A Comprehensive Analysis of Classical, Machine Learning, and Deep Learning Based Methods," IEEE Access, vol. 10, pp. 38999-39044 April 2022.

[2] M. R. Hossain, and M. M. Hoque, "Towards Bengali Word Embedding: Corpus Creation, Intrinsic and Extrinsic Evaluations," Preprints 2020, 2020120600 (doi: 10.20944/preprints202012.0600.v1).

[3] M. R. Hossain, M. M. Hoque, N. Siddique, I. H. Sarkar, "Bengali text document categorization based on very deep convolution neural network," Expert Systems with Applications. England, vol. 184, pp. 115394, December 2021.

[4] E. A. Emon, S. Rahman, J. Banarjee, A. K. Das, T. Mittra, "A deep learning approach to detect abusive bengali text," In2019 7th International

Conference on Smart Computing & Communications (ICSCC). Malaysia, pp. 1-5, June 2019.

[5] T. T. Mayeesha, A. M, Sarwar, R. M. Rahman, "Deep learning based question answering system in Bengali," Journal of Information and Telecommunication. England, vol. 5, no. 2, pp. 145-178, April 2021.

[6] M. Rahman, S. Haque, and Z. R. Saurav, "Identifying and categorizing opinions expressed in Bangla sentences using deep learning technique," International Journal of Computer Applications., vol. 176, pp. 8887, April 2020.

[7] W. Akanda, and A. Uddin, "Multi-Label Bengali article classification using ML-KNN algorithm and Neural Network," 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD). Bangladesh, pp. 466-471, April 2021.

[8] M. R. Amin, and M. Chakraborty, "Algorithm for Bengali keyword extraction," In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP). Bangladesh, pp. 1-5, September 2018.

[9] U. Brunner, and K. Stockinger, "Valuenet: A natural language-to-sql system that learns from database information." In 2021 IEEE 37th International Conference on Data Engineering (ICDE). Greece, pp. 2177-2182, April 2021.

[10] S. S. Badhya, A. Prasad, S. Rohan, Y. S. Yashwanth, N. Deepamala, and G. Shobha. "Natural language to structured query language using elasticsearch for descriptive columns," In 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS). India, vol. 4, pp. 1-5, December 2019.

[11] A. Das, and R. C. Balabantaray, "MyNLIDB: a natural language interface to database," In 2019 International Conference on Information Technology (ICIT). India, pp. 234-238, December 2019.

[12] M. Kaufmann, G. Stechschulte, and A. Huber, "Efficient and Accurate In-Database Machine Learning with SQL Code Generation in Python," arXiv preprint arXiv:2104.03224 (2021).

[13] T. Bai, Y. Ge, S. Guo, Z. Zhang, and L. Gong, "Enhanced natural language interface for web-based information retrieval," IEEE Access, vol. 9, pp. 4233-4241, December 2020.

[14] M. Eminağaoğlu, and Y. Gökşen, "A new similarity measure for document classification and text mining," KnE Social Sciences, pp. 353-366, January 2020.

[15] Y. Tang, "Research on Word Vector Training Method Based on Improved Skip-Gram Algorithm," Advances in Multimedia, pp. 1-8, February 2022.

[16] A. A. A. Rafat, M. Salehin, F. R. Khan, S. A. Hossain, and S. Abujar, "Vector Representation of Bengali Word Using Various Word Embedding Model," In 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART). India, pp. 27-30, November 2019.

[17] S. H. Sumit, M. Z. Hossan, T. A. Muntasir, and T. Sourov, "Exploring word embedding for bangla sentiment analysis," In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP). Bangladesh, pp. 1-5, September 2018.

[18] P. S. Kambali, S. Suri, and B. M. Sagar. "Distributed Representation of Words in Vector Space for Kannada Language," In 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS). India, pp. 54-58, December 2018.

[19] P. K. Saha, A. Das Mou, and T. Mittra. "A Bangla Word Sense Disambiguation Technique using Minimum Edit Distance Algorithm and Cosine Distance," In 2019 23rd International Computer Science and Engineering Conference (ICSEC). Thailand, pp. 1-6, November 2019.

[20] Q. Du, N. Li, W. Liu, D. Sun, S. Yang, and F. Yue. "A Topic Recognition Method of News Text Based on Word Embedding Enhancement," Computational Intelligence and Neuroscience, pp. 1-15, February 2022.

[21] Y. Hu, H. He, Z. Chen, Q. Zhu, and C. Zheng. "A Unified Model Using Distantly Supervised Data and Cross-Domain Data in NER," Computational Intelligence and Neuroscience, pp. 1-11, May 2022.

[22] B. Wang, and Z. Ou. "Learning neural trans-dimensional random field language models with noise-contrastive estimation," In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Canada, pp. 6134-6138, April 2018.

[23] A. Egitmen, I. Bulut, R. Aygun, A. B. Gunduz, O. Seyrekbasan, and A. G. Yavuz. "Combat mobile evasive malware via skip-gram-based malware detection," Security and Communication Networks, pp. 1-10, April 2020.

[24] A. O. Nicholas, I. O. Ndaman, S. Misra, O. O. Abayomi-Alli, and R. Damaševičius. "Text messaging-based medical diagnosis using natural language processing and fuzzy logic," Journal of Healthcare Engineering, pp. 1-14, September 2020.

[25] C. Zhang, X. Liu, and D. Biś, "An analysis on the learning rules of the skip-gram model," In 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 1-8, July 2019.

[26] O. Chehab, A. Gramfort, and A. Hyvärinen, "The optimal noise in noise-contrastive learning is not what you think," In Uncertainty in Artificial Intelligence, pp. 307-316, August 2022.

[27] S. Mohamad, H. Alamri, and A. Bouchachia, "Scaling up stochastic gradient descent for non-convex optimisation," Machine Learning, pp. 1-41, October 2022.

[28] M. Arefin, K. M. Hossen, and M. N, Uddin. "Natural Language Query to SQL Conversion Using Machine Learning Approach," In 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI). Bangladesh, pp. 1-6, December 2021.

[29] J. Liu, Q. Cui, H. Cao, T. Shi, and M. Zhou. "Auto-conversion from Natural Language to Structured Query Language using Neural Networks Embedded with Pre-training and Fine-tuning Mechanism," In 2020 Chinese Automation Congress (CAC). China, pp. 6651-6654, November 2020.

[30] H. Sanyal, S. Shukla, and R. Agrawal. "Natural Language Processing Technique for Generation of SQL Queries Dynamically," In 2021 6th International Conference for Convergence in Technology (I2CT). India, pp. 1-6, April 2021.

[31] C. Sugandhika, and S. Ahangama. "Heuristics-Based SQL Query Generation Engine," In 2021 6th International Conference on Information Technology Research (ICITR). Sri Lanka, pp. 1-7, December 2021.

[32] D. Pal, H. Sharma, and K. Chaudhuri. "Data Agnostic RoBERTa-based Natural Language to SQL Query Generation," In 2021 6th International Conference for Convergence in Technology (I2CT). India, pp. 1-5, April 2021.

[33] S. Huo, T. Ma, J. Chen, M. Chang, L. Wu, and M. J. Witbrock. "Graph enhanced cross-domain text-to-sql generation," In Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13), pp. 159-163, November 2019.

[34] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. "Distributed representations of words and phrases and their compositionality," Advances in neural information processing systems, vol. 26, 2013.

[35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M Isard, M. Kudlur "{TensorFlow}: a system for {Large-Scale} machine learning," In12th USENIX symposium on operating systems design and implementation (OSDI 16), pp. 265-283, 201.

[36] A. Mnih, and Y. W. The, "A fast and simple algorithm for training neural probabilistic language models," arXiv preprint arXiv:1206.6426 (2012).

[37] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747 (2016).

[38] K. P. Mandal, P. Mukherjee, A. Chattopadhyay, B. Chakraborty, "XBLQPS: An Extended Bengali Language Query Processing System for e-Healthcare Domain," vol. 13, pp. 502-516, 2022.

Author's Biography



Kailash Pati Mandal obtained his Bachelor of Technology in Computer Science and Engineering from Bengal College of Engineering and Technology, Durgapur, India in 2006. In 2011, he received Master's degree in Computer Science and Engineering from the Jadavpur University, Kolkata, India. He is currently a part-time Researcher in Computer Science and Engineering in the field of natural language processing at the National Institute of Technology (NIT), Durgapur, India.



Dr. Prasenjit Mukherjee obtained his PhD in Computer Science and Engineering from National Institute of Technology (NIT), Durgapur, India under Visvesvaraya PhD scheme Program, Ministry of Electronics and Information Technology (MeitY), Govt. of India. His research interests include machine learning, deep learning, natural language processing, knowledge engineering, and database management systems. He has >15 international publications.



Devraj Vishnu received the B. Tech degree in Computer Science and Engineering from Bengal College of Engineering and Technology, Durgapur, India, in 2013. In 2017, he received the M. Tech. degree in Printing Engineering & Graphic Communication from Jadavpur University, Kolkata, India. He is presently working as an Independent Researcher, Kolkata, West Bengal, India.



Dr. Baisakhi Chakraborty obtained her PhD in Computer Science and Engineering from National Institute of Technology (NIT), Durgapur, India in 2011. Her research interests include knowledge systems, knowledge engineering and management, database systems, data mining, natural language processing, and software engineering. She has several researchers under her direction. She has >30 international publications. She has a decade of industry experience and 17 years of academic experience.



Dr. Tanupriya Choudhury completed his undergraduate studies in Computer Science and Engineering at the West Bengal University of Technology in Kolkata (2004–2008), India, followed by a master's degree in the same field from Dr. M.G.R University in Chennai, India (2008–2010). In 2016, he successfully obtained his PhD degree from Jagannath University, Jaipur. With a total of 15 years of experience in both teaching and research, Dr. Choudhury holds the position of visiting professor at Daffodil International University, Bangladesh. Currently, he is working as a professor, Graphic Era Deemed to be University, Dehradun, India.

Prior to this role, he served Symbiosis International University, Pune, Graphic Era Hill University Dehradun (research professor), UPES Dehradun (professor and research head of informatics), Amity University Noida (assistant professor Grade 3 and international depatment head), and other prestigious academic institutions (Dronacharya College of Engineering Gurgaon, Lingaya's University Faridabad, Babu Banarsi Das Institute of Technology, Ghaziabad, Syscon Solutions Pvt.

Ltd. Kolkata, etc.). Recently, he was recognized for his outstanding contributions to education with the Global Outreach Education Award for Excellence in Best Young Researcher Award at GOECA 2018.

His areas of expertise encompass human computing, soft computing, cloud computing, data mining, among others. Notably accomplished within his field thus far is filing 25 patents and securing copyrights for 16 software programs from the Ministry of Human Resource Development (MHRD). He has actively participated as an attendee or speaker at numerous national and international conferences across India and abroad.

As an active professional within the technical community; Dr. Choudhury holds lifetime membership with the International Engineering & Technology Association (IETA) along with being affiliated with the Institute of Electrical and Electronics Engineers (IEEE), IET (UK) (Institution of Engineering & Technology UK), and other reputable technical societies. Additionally, he is associated with corporate entities and serves as a technical adviser for Deetya Soft Pvt. Ltd., Noida, IVRGURU, and Mydigital360. He is also serving an editor's in reputed journals.

He currently serves as the honorary secretary in the Indian Engineering Teacher's Association (IETA), India, alongside his role as the senior advisor position in INDO-UK Confederation of Science, Technology and Research Ltd., London, UK, and International Association of Professional and Fellow Engineers, Delaware, USA.



Dr. Pradeep Kumar Arya, a distinguished academician and researcher, holds a Ph.D. in Computer Science and Engineering from Anna University Chennai. Currently, an Associate Professor at *Bennett University, Greater Noida*, Uttar Pradesh, India. His expertise spans Information Security, Cloud Computing, and Software Engineering. With a prolific teaching and research career, Dr. Arya has guided numerous postgraduate and undergraduate students and contributed significantly to academia. His research focuses on enhancing security in cloud applications, addressing Authentication, Security in AIML applications and IoT security. He also holds patents for innovative inventions. He actively participates in workshops, symposiums, and conferences, leaving a lasting impact on computer science and engineering advancement.