



International Journal for Innovative Engineering and Management Research

A Peer Reviewed Open Access International Journal

www.ijiemr.org

COPY RIGHT



ELSEVIER
SSRN

2022 IJIEMR. Personal use of this material is permitted. Permission from IJIEMR must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. No Reprint should be done to this paper, all copy right is authenticated to Paper Authors

IJIEMR Transactions, online available on 26th Apr 2022. Link

[:http://www.ijiemr.org/downloads.php?vol=Volume-11&issue= Spl Issue 04](http://www.ijiemr.org/downloads.php?vol=Volume-11&issue= Spl Issue 04)

DOI: 10.48047/IJIEMR/V11/SPL ISSUE 04/07

Title **CHATBOT USING FINE TUNED RANDOM FOREST**

Volume 11, SPL ISSUE 04, Pages: 67-76

Paper Authors

Dr. Sk. Akbar , G. Vara Lakshmi, K. J Harsha Vardhan, M. Yesaswini, V. Prudhvi Charan , M. Deepikeswari



USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per **UGC Guidelines** We Are Providing A Electronic Bar Code

CHATBOT USING FINE TUNED RANDOM FOREST

Dr. Sk. Akbar¹, G. Vara Lakshmi², K. J Harsha Vardhan³, M. Yesaswini⁴, V. Prudhvi Charan⁵, M. Deepikeswari⁶

¹Professor CSE, PSCMR College of Engineering & Technology, Vijayawada, Andhra Pradesh.

^{2,3,4,5,6} Student, CSE, PSCMR CET, Vijayawada, Andhra Pradesh.

¹dr.akbar@pscmr.ac.in, ²varalakshmigunti21@gmail.com, ³harsha.kollipara2791@gmail.com,
⁴mallyesawini@gmail.com, ⁵prudhvicharanv@gmail.com, ⁶deepikamanjula14@gmail.com

ABSTRACT:

The Answering System is a type of information retrieval system in which a direct response is anticipated in response to a submitted query, as opposed to a list of references that could contain the answers to the questions. It is a piece of hardware that allows for communication between humans and machines. The quality assurance mechanisms used in natural language processing are intended to provide students with correct solutions to their questions. This article gives an overview of the many quality assurance (QA) solutions that are available. QA systems may be broken down into a few different categories, including text-based QA systems, factoid-based QA systems, Web-based QA systems, Information Retrieval or Extraction-based QA systems, Restricted Domain QA systems, and rule-based QA systems. This work analyses further a comparative assessment of these models for various sorts of questioners, which resulted in a breakthrough for new research pathways in this topic.

Keywords: Machine Learning, Natural Language Processing, Questioning Answering System, Query, retrieval, response, Naïve Bayes, SVM, Random Forest, etc.

1. INTRODUCTION

Unlike the majority of information retrieval systems, QA systems strive to retrieve point-by-point responses as opposed to a deluge of documents or even matching sections. The most difficult aspect of a question-answering system is providing reliable responses from the vast amount of web-based data. The processing of time-based data to respond to temporal inquiries is still a challenge. This study focuses on many types of quality assurance (QA) systems.

Research on quality assurance aims to cover a wide range of question types, including fact,

list, definition, how, why, hypothetical and semantically constrained among others. The question answering system may, in general, be broken down into two categories: the closed domain question answering systems and the open domain question answering systems. It is possible that answering questions posed within a closed domain would appear to be an easier task to complete due to the fact that NLP systems may employ domain-specific knowledge that is often defined in ontologies. An alternative usage of the term "closed domain" might refer to a circumstance in which only a select few

varieties of questions are permitted, such as searches looking for descriptive information as opposed to information on how to carry out a certain procedure. When it comes to providing responses to questions about virtually any topic, open-domain question answering can only rely on broad ontologies and global knowledge. On the other hand, these systems often have a substantially larger amount of data from which to derive the answer.

1.1 Problem Statement

Question Answering (QA) is the most common way of separating brief, pertinent literary reactions to normal language requests. As a subset of QA, titbit QA accentuates questions whose answers are syntactic as well as semantic things, for example, association and individual names. Various significant certifiable purposes for QA frameworks exist, for example, internet searcher updates and robotized client support.

The present status of the workmanship in QA innovation blends AI with language data encoded by human experts as rules or heuristics in most of the parts of QA frameworks (Pasca, 2003). At the point when an elevated degree of inclusion is looked for, the best downside of these advancements is their restrictively costly turn of events and change costs.

Question Processing, which distinguishes the kind of the offered conversation starter, Passage Retrieval (PR), which removes few significant sections from the basic discourse records, and Answer Extraction (AE), which concentrates and positions precise responses from the recently recovered entries, contain the common design of the QA framework

presented. Handling of Queries: The QP part decides the sort of info inquiries by planning them to a two-level scientific categorization including six inquiry classes and 53 subclasses.

Recovery of a section - The PR comprises two fundamental stages: (a) in the main stage, all constant question words are arranged in plunging need request, and (b) in the subsequent advance, the rundown of watchwords utilized for recovery and their closeness is progressively different until a sufficient number of entries are recuperated.

Answer Extraction - The Answer Extraction (AE) part distinguishes competitor answers from the significant section set and concentrates the answer(s) probably going to resolve the client's inquiry.

1.2 Ambition

Our ambition associated with this project is to build an efficient question answering system with various ML techniques and reproduce a comparative study on which will be the best one for the same.

1.3 Objectives

Now a days Chatbots plays an important role in the various sectors. Consumers are demanding round-the-clock service for assistance in areas ranging from banking and finance, to health and wellness. Because of this demand, chatbots are increasing in popularity among business and consumers alike. The Objective of our project is to review various ML techniques and comprehend their Limitations and Advantages. And to get a thorough list of all the Natural Language Processing methods. Finally, our main motive is to build an

efficient system and choose the best algorithms for the QA model.

1.4 Significance of the Project

Web search tools give a focus on the rundown of applicable reports in light of client entered catchphrases because of an assortment of elements, including notoriety estimations, watchword coordinating, archive access frequencies, and so on. Clients should survey each record independently to get the required data (Ferret et al., 2001); this makes data recovery a tedious activity. An internet searcher ought to, preferably, return a few significant, short sentences as answers along with their separate web joins. Since the 1960s, various QASs have been created (Androutsopoulos et al., 1995, Kolomiyets, 2011). Current QASs look to respond to questions presented by clients in normal dialects by gaining and breaking down information from numerous information sources, including the semantic web (Vanessa, 2011, Dwivedi, 2013, Suresh Kumar and Zayaraz, 2014). The configuration of reactions will similarly move from basic text to mixed media (Voorhees and Weishedel, 2000). The quantity of QASs laid out since the 1960s that survey different subjects, information sources, question types, answer designs, and so forth is unnecessary. To assess the progress of these QASs and their capacity to meet current and future necessities, an exhaustive assessment of all QASs is required. In this work, we arrange QASs in light of qualities, for example, application regions, questions, information sources, matching capacities, and answers. We survey the writing on QASs sorted by every measure and recognize future examinations open doors in this field.

2. LITERATURE SURVEY

In this section, we discuss the evolution of QASs from the 1960s to the present day. In the fifth generation of computer programming language, the plan to construct systems capable of handling natural language questions was initiated.

2.1 Related Works

NLIDB is a framework that gives the office to clients to pose inquiries in their normal dialects for getting data from data sets (Androutsopoulos et al., 1995). It facilitates human PC association as clients need not learn formal dialects like SQL, Prolog, Lisp, and so forth for submitting inputs. Green et al. (1961) propose BASEBALL, a QAS that gives data related to a baseball association played in America during a specific season. This framework gives replies to questions connected with dates, area, and so on. Woods (1973) propose LUNAR, a QAS that gives data about soil tests taken from Apollo lunar investigation. These frameworks change clients' inquiries into data set questions through plain example matching principles and lastly produce replies. These plain examples matching guidelines use restricted punctuations, permanently set up information, and planning rules which rely on application areas. As a characteristic language upholds rewording, handling regular language inquiries through design matching is anything but a plausible arrangement. Both BASEBALL and LUNAR frameworks produce great outcomes, however, they have a restricted storehouse of data connected with their application areas.

2.2 Insights from other researchers

The examination of open space questions addressing unstructured information sources was launched by the TREC Evaluation crusade which is occurring consistently beginning around 1999 (Voorhees, 2001, Voorhees, 2004, Voorhees and Weishedel, 2000). The principal TREC assessment crusade gives a rundown of 200 inquiries and an archive assortment. The responses were known to be available in the assortments. The greatest lengths of answers were permitted to be 50 or 250 characters. Frameworks were requested to give 5 positioned records from replies. In the following effort, TREC-9 held in 2000, the number of inquiries and size of archive assortments was expanded. In TREC-10 out of 2001, another intricacy regarding replies, i.e., answer approval task was incorporated as there was no affirmation, all things considered, to be available in the archive assortments. The lengths of answers were diminished to 50 words. In TREC-11, held in 2002, frameworks were supposed to offer precise short responses to the inquiries. In TREC from 2002 to 2007, the rundown of inquiries, definition questions, and tidbit questions was remembered for the assessment crusades. In TREC 2005, there were a bunch of 75 points that contains different kinds of inquiries (rundown, tidbit, or others). Transient inquiries were added to TREC 2005 and TREC 2006. In TREC 2007, report assortments included blog assortments. TREC rivalries progress with expanding size and intricacy of record assortments; expanding intricacy of inquiries; and expanding intricacy of answer assessment systems.

3. PROPOSED SYSTEM

In this section, we shall deeply discuss the methodology and procedure that we have followed to build the QA system.

3.1 System Architecture

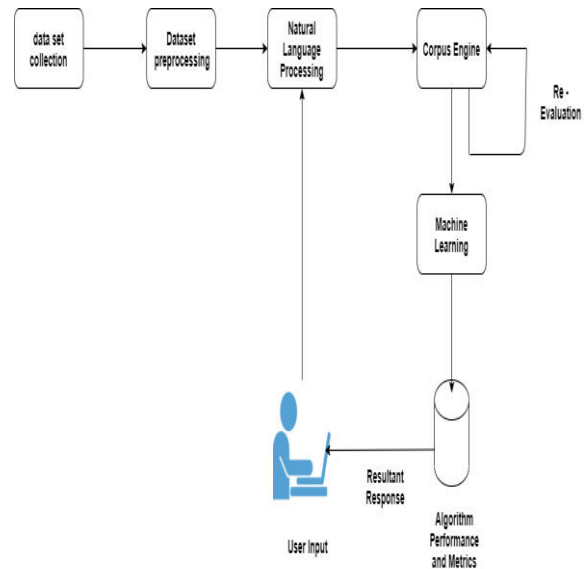


Figure 1: System Architecture

3.2 Algorithm Insights

In this project, we have utilized the multiple machine learning approaches of Support Vector Machines, Random Forest, and Naive Bayes models together with Tensor Flow in order to replicate a Question Answer Machine Natural Language Processing system.

The Random Forest Algorithm offers a number of benefits, one of the most notable of which is that it reduces the risk of overfitting as well as the amount of training time that is required. In addition to that, it offers an exceptionally high degree of accuracy. The Random Forest approach works quickly on large databases and gives predictions that are exceptionally accurate. It

does this by providing approximations for data that is absent. Because the random forest uses a combination of different trees to make its predictions about the dataset's category, some decision trees could make accurate forecasts while others might not.

SVM works sensibly actually when there is an unmistakable division between classes. SVM performs better in high-layered spaces. In circumstances where the quantity of aspects surpasses the number of tests, SVM is helpful. SVM is generally effective with memory. There are a few arrangement calculations utilized in AI, be that as it may, SVM is better than most of them since it creates more exact outcomes. It can execute similar tasks in the n-layered space of the choice limit partitioning the two classes as it can in the n-layered space of the limit isolating the two classes. Huge informational indexes are not appropriate for the SVM strategy. SVM performs ineffectively when the informational collection contains more clamor, for example whenever target classes cross over. In circumstances where the quantity of elements per information point surpasses the quantity of preparing information tests, the SVM will perform inadequately.

Bayes performs very well while the preparation information does exclude every single imaginable result, consequently, it tends to be profoundly successful with little amounts of data. Choice trees perform preferable with additional information over Naive Bayes. Bayes is generally utilized in mechanical technology and PC vision, where it performs outstandingly.

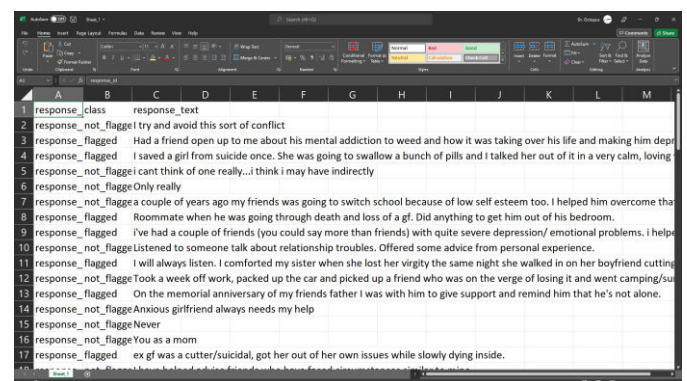
It is clear to execute. It expects undeniably less preparation information. It handles both

discrete and nonstop information. It scales well concerning the number of indicators and data of interest. It is speedy and can be utilized to make expectations progressively.

Bayes is a classifier that anxiously learns and is essentially quicker than K-NN. Along these lines, it very well may be used for continuous expectation. Ordinarily, the Naive Bayes classifier is utilized for email spam separation. It produces likelihood for each class in light of probabilistic assessment.

3.3 Importing Packages & Spreadsheets

In our very first step, we import all the necessary packages of pandas, NumPy, seaborn, Keras, and Tensor Flow. After the successful imports of all the needed libraries and packages, we proceed to the next step of importing the spreadsheets in CSV formats. We have two data files, where we named each as chatbot and resume file, the chatbot head contains Flagged or not flagged class, response text whereas the resume head contains id, Flag, Not Flagged class along with the resume text of the machine.



response_class	response_text
response_not_flagged	I try and avoid this sort of conflict
response_flagged	Had a friend open up to me about his mental addiction to weed and how it was taking over his life and making him depressed
response_flagged	I saved a girl from suicide once. She was going to swallow a bunch of pills and I talked her out of it in a very calm, loving way
response_not_flagged	I cant think of one really...i think i may have indirectly
response_not_flagged	Only really
response_not_flagged	a couple of years ago my friends was going to switch school because of low self esteem too. I helped him overcome that
response_flagged	Roommate when he was going through death and loss of a gf. Did anything to get him out of his bedroom.
response_flagged	I've had a couple of friends (you could say more than friends) with quite severe depression/ emotional problems. I helped them
response_not_flagged	Listened to someone talk about relationship troubles. Offered some advice from personal experience.
response_flagged	I will always listen. I comforted my sister when she lost her virginity the same night she walked in on her boyfriend cutting himself
response_not_flagged	Took a week off work, packed up the car and picked up a friend who was on the verge of losing it and went camping/suicide
response_flagged	On the memorial anniversary of my friends father I was with him to give support and remind him that he's not alone.
response_not_flagged	Anxious girlfriend always needs my help
response_not_flagged	Never
response_not_flagged	You as a mom
response_flagged	ex gf was a cutter/suicidal, got her out of her own issues while slowly dying inside.

Figure 2: Sample Dataset

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline

np.random.seed(1337)

from keras.models import Sequential
from keras.layers import Embedding, Dense, LSTM, BatchNormalization
from keras.layers import SpatialDropout1D
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences

Using TensorFlow backend.
```

Figure 3: Code Snippet of Packages Import

response_id	class	response_text
0	not_flagged	I try and avoid this sort of conflict
1	flagged	Had a friend open up to me about his mental ad...
2	flagged	I saved a girl from suicide once. She was goin...
3	not_flagged	I cant think of one really...I think I may hav...
4	not_flagged	Only really one friend who doesnt fit into th...

resume_id	class	resume_text
0	not_flagged	\rCustomer Service Supervisor/Tier - Isabella ...
1	not_flagged	\rEngineer / Scientist - IBM Microelectronics ...
2	not_flagged	\rLTS Software Engineer Computational Lithogra...
3	not_flagged	TUTORVWilliston VT - Email me on Indeed: ind...
4	flagged	\rIndependent Consultant - Self-employed\rBurl...

Figure 4: Code Snippet of Data Import

3.4 Vectorization in Natural Language Processing

Word Embeddings are a method that converts a string of text into a collection of real numbers in the form of a vector, which is required for the application of AI and deep learning techniques to the processing of natural language text in order to extract meaningful information from a particular word. Word Embeddings, also known as Word Vectorization, is a technique that is used in natural language processing (NLP) to convert words or expressions from jargon into a vector of real numbers. These numbers are then applied to the problem of determining word expectations as well as word similitudes and semantics. The conversion of words into numbers is most

frequently done through the process of vectorization.

Normal Language Processing requires the transformation of text/strings to genuine numbers, known as word embeddings or word vectorization. Whenever words are changed over completely to vectors, Cosine comparability is used to fulfill most of the purpose cases for Natural Language Processing, Document bunching, and Text orders. This technique predicts words in light of the sentence set. Cosine Similarity — "As the point diminishes, the likeness increments."

Word2Vec, Fast text, and Glove are the most notable plans for changing over word vectors and utilizing cosine closeness for word similitude highlights. NNLM and RNNLM beat for the gigantic corpus of words. Be that as it may, calculation intricacy is a tremendous expense. Word2Vec utilizes CBOW and Skip-gram engineering to augment precision and limit calculation intricacy to conquer calculation trouble. CBOW engineering expects the current word contingent upon its unique circumstance. The design of the skip-gram predicts encompassing words in light of the current word.

```
count_vect = CountVectorizer()

x = chatbot['response_text']
y = chatbot['class']
x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=1)
X_train_counts = count_vect.fit_transform(x_train)
X_test_counts = count_vect.transform(x_test)
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
X_test_tfidf = tfidf_transformer.fit_transform(X_test_counts)

print(X_train_tfidf.shape)
print(X_test_tfidf.shape)

(68, 572)
(20, 572)
```

Figure 5: Vectorization of Data

3.5 Built Models Results

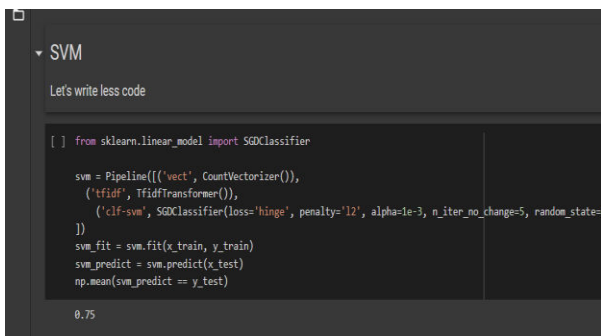
In this section, we will see the results and accuracies obtained from the above 3 models of Machine Learning techniques after parsing through the Vectorization phase of NLP.

Random Forest

```
rf = RandomForestClassifier(max_depth=10,max_features=10)
rf.fit(x_train_dtm,y_train)
rf_predict = rf.predict(x_test_dtm)
metrics.accuracy_score(y_test,rf_predict)

0.80000000000000004
```

Figure 6: Fine Tuned Random Forest Predictions



```
from sklearn.linear_model import SGDClassifier

svm = Pipeline([('vect', CountVectorizer()),
               ('tfidf', TfidfTransformer()),
               ('clf-svm', SGDClassifier(loss='hinge', penalty='l2', alpha=1e-3, n_iter_no_change=5, random_state=4))
              ])
svm_fit = svm.fit(x_train, y_train)
svm_predict = svm.predict(x_test)
np.mean(svm_predict == y_test)

0.75
```

Figure 7: Enhanced SVM Predictions

Naive bayesian

```
In [22]: from sklearn.naive_bayes import MultinomialNB

naive = MultinomialNB().fit(X_train_tfidf, y_train)
predicted = naive.predict(X_test_tfidf)
np.mean(predicted == y_test)

Out[22]: 0.7
```

Figure 8: Naïve bayes Predictions

After the fit function, we have used functional Max no of words to be 50 thousand only, max no of words in each compliant, sequence length to be 300, embedding to be 100. Post this we truncate and pad the sequence to have the string representation of each size, and convert the label to categorical values.

Now we build the sequential model and add 2 dense layers with softmax activation function and enable adam optimizer with accuracy metrics and categorical cross-entropy loss. We have trained on 83 Samples and Validated 10 samples. We have gotten 8 epochs whose results are as below-

```
Epoch 1/8 - 2s 26ms/step - loss: 0.6865 - accuracy: 0.5904 - val_loss: 0.6699 - val_accuracy: 0.8000
Epoch 2/8 - 1s 11ms/step - loss: 0.6615 - accuracy: 0.7590 - val_loss: 0.6347 - val_accuracy: 0.8000
Epoch 3/8 - 1s 12ms/step - loss: 0.6174 - accuracy: 0.7590 - val_loss: 0.5620 - val_accuracy: 0.8000
Epoch 4/8 - 1s 11ms/step - loss: 0.5250 - accuracy: 0.7590 - val_loss: 0.4854 - val_accuracy: 0.8000
Epoch 5/8 - 1s 13ms/step - loss: 0.5150 - accuracy: 0.7590 - val_loss: 0.4831 - val_accuracy: 0.8000
Epoch 6/8 - 1s 12ms/step - loss: 0.4686 - accuracy: 0.7590 - val_loss: 0.4967 - val_accuracy: 0.8000
Epoch 7/8 - 1s 11ms/step - loss: 0.4356 - accuracy: 0.7590 - val_loss: 0.4833 - val_accuracy: 0.8000
Epoch 8/8 - 1s 11ms/step - loss: 0.3763 - accuracy: 0.7590 - val_loss: 0.4606 - val_accuracy: 0.8000
```

4. CONCLUSION AND FUTURE SCOPE

From the above epochs' description, we can tell that the best accuracy was achieved in the least loss one i.e., the last epoch. For the test set, the loss has derivate to be 0.68 and accuracy to be 0.80.

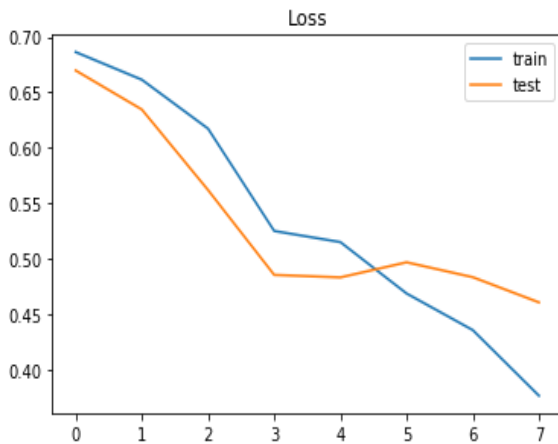


Figure 9: Train & Test Set Loss Graph

From the above graph, we can understand that the loss for both the test set and the train set has declined over time for all the 3 models of Support Vector Machines, Random Forest and Naïve Bayes. Hence, our model has been successful in making.

Likewise with whatever other overview, this paper offers assistance to established researchers. It blended and imaginatively organized late exploration discoveries, along these lines incorporating and improving the field of inquiry addressing. It featured the grouping of the ongoing writing, the development of a point of view regarding the matter, and the assessment of patterns. Since a review can exclude all or even most of past examinations, this study just incorporated crafted by the most productive and referred to creators in the QA field. Likewise, because logical exploration is a moderate, consistent, and collective undertaking, this review included research with unobtrusive defects to exhibit how these requirements were distinguished, defied, and tended to by different analysts.

5. REFERENCES

- [1] Alzubi, J.A., Jain, R., Singh, A. *et al.* COBERT: COVID-19 Question Answering System Using BERT. *Arab J Sci Eng* (2021). <https://doi.org/10.1007/s13369-021-05810-5>
- [2] D. V. Vekariya and N. R. Limbasiya, "A Novel Approach for Semantic Similarity Measurement for High Quality Answer Selection in Question Answering using Deep Learning Methods," *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 518-522, doi: 10.1109/ICACCS48705.2020.9074471.
- [3] Mutabazi, E.; Ni, J.; Tang, G.; Cao, W. A Review on Medical Textual Question Answering Systems Based on Deep Learning Approaches. *Appl. Sci.* 2021, 11, 5456. <https://doi.org/10.3390/app11125456>.
- [4] M. R. Bhuiyan, A. K. M. Masum, M. Abdullahil-Oaphy, S. A. Hossain and S. Abujar, "An Approach for Bengali Automatic Question Answering System using Attention Mechanism," *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1-5, doi: 10.1109/ICCCNT49239.2020.9225264.
- [5] K. Moholkar and S. H. Patil, "Multiple Choice Question Answer System using Ensemble Deep Neural Network," *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2020, pp. 762-766, doi: 10.1109/ICIMIA48430.2020.9074855.

- [6] L. Hirschman and R. Gaizauskas, "Natural language question answering: the view from here," *Natural Language Engineering*, vol. 7, no. 4, pp. 275-300, 2001.
- [7] D. Zhang and W. Lee, "A Web-based Question Answering System," Massachusetts Institute of Technology (DSpace@MIT), 2003.
- [8] P. Banerjee and H. Han, "Drexel at TREC 2007: Question Answering," in *Proceedings of the Sixteenth Text Retrieval Conference (TREC 2007)*, 2007.
- [9] M. M. Sakura, M. M. Kouta, and A. M. N. Allam, "Automated Construction of Arabic-English Parallel Corpus," *Journal of the Advances in Computer Science*, vol. 3, 2009.
- [10] M. Ramprasath and S. Hariharan, "A Survey on Question Answering System," *International Journal of Research and Reviews in Information Sciences (IRIS)*, pp. 171-179, 2012.
- [11] S. Stoyanchev, Y. Song and W. Lahti, "Exact phrases in information retrieval for question answering," in *Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, 2008.
- [12] M. M. Sakura, M. M. Kouta, and A. M. N. Allam, "Weighting Query Terms Using Wordnet Ontology," *International Journal of Computer Science and Network Security*, vol. 9, no. 4, pp. 349-358, 2009.
- [13] E. Voorhees, "Overview of the TREC 2002 Question Answering Track," in *Proceedings of the Text Retrieval Conference (TREC 2002)*, 2002.
- [14] B. F. Green, A. K. Wolf, C. Chomsky, and K. Laughery, "BASEBALL: An automatic question answerer," in *Proceedings of Western Joint IRE-AIEE-ACM Computing Conference*, Los Angeles, 1961.
- [15] W. Woods, "Progress in Natural Language Understanding: An Application to Lunar Geology," in *Proceedings of the National Conference of the American Federation of Information Processing Societies*, 1973.
- [16] C. Paris, "Towards More Graceful Interaction: A Survey of Question-Answering Programs," *Columbia University Computer Science Technical Reports*, 1985.
- [17] W. G. Lehnert, *The Process of Question Answering - A Computer Simulation of Cognition*, Yale University, 1977.
- [18] D. Radev, W. Fan, H. Qi, H. Wu and A. Grewal, "Probabilistic Question Answering on the Web," *Journal of the American Society for Information Science and Technology*, vol. 56, no. 6, pp. 571-583, 2005.
- [19] S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus and P. Morarescu, "FALCON: Boosting Knowledge for Answer Engines," in *Proceedings of the Ninth Text Retrieval Conference (TREC9)*, 2000.
- [20] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, 2002.
- [21] R. Gaizauskas and K. Humphreys, "A Combined IR/NLP Approach to Question Answering Against Large Text Collections," in *Proceedings of the 6th Content-based Multimedia Information Access (RIAO-2000)*, 2000.

[22] M. Kangavari, S. Ghandchi, and M. Golpour, "Information Retrieval: Improving Question Answering Systems by Query Reformulation and Answer Validation," World Academy of Science, Engineering and Technology, pp. 303-310, 2008.

[23] D. Ravichandran and E. Hovy, "Learning Surface Text Patterns for a Question Answering System," in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002.

[24] D. Zhang and W. Lee, "Question Classification using Support Vector Machines," in Proceedings of the 26th Annual International ACM SIGIR Conference, 2003.