
Bayesian Probability and Tanimoto Based Recurrent Neural Network for Question Answering System

Veeraraghavan Jagannathan

*Associate Professor, Department of Computer Science and Engineering, Sri Vasavi Engineering College (Autonomous), Pedatadepalli, Tadepalligudem-534101. Andhra Pradesh, India
E-mail: vjveeraraghavan@gmail.com*

Received 23 March 2020; Accepted 31 January 2021;
Publication 02 June 2021

Abstract

Question Answering (QA) has become one of the most significant information retrieval applications. Despite that, most of the question answering system focused to increase the user experience in finding the relevant result. Due to the continuous increase of web content, retrieving the relevant result faces a challenging issue in the Question Answering System (QAS). Thus, an effective Question Classification (QC), and retrieval approach named Bayesian probability and Tanimoto-based Recurrent Neural Network (RNN) are proposed in this research to differentiate the types of questions more efficiently. This research presented an analysis of different types of questions with respect to the grammatical structures. Various patterns are identified from the questions and the RNN classifier is used to classify the questions. The results obtained by the proposed Bayesian probability and Tanimoto-based RNN showed that the syntactic categories related to the domain-specific types of proper nouns, numeral numbers, and the common nouns enable the RNN classifier to reveal better result for different types of questions. However, the proposed approach obtained better performance in terms of precision, recall, and F-measure with the values of 90.14, 86.301, and 90.936 using dataset-2.

Journal of Web Engineering, Vol. 20_3, 903–934.
doi: 10.13052/jwe1540-9589.20315
© 2021 River Publishers

Keywords: Question answering system, question classification, recurrent neural network, Bayesian probability, machine learning.

1 Introduction

Search engines are used for finding the answers to the questions in daily life. However, the search engine requires the user to identify the effective searching keywords for the questions, without spending a long time for the users to search the answer for the questions [7, 13, 29]. Moreover, the search engine offers the documents in a ranking order that includes only partial content, which makes it more complex for the user to find interesting documents. As, the questions created by the user are typically heterogeneous, personal, open-ended, and extremely specific, the search engines are widely not enough to identify the single web page, which contains the direct answer for the user question [7, 14]. However, real humans are better in answering and understanding than the machine [7, 15]. There exist some emerging trends to the websites, such as Google Answer [11], Naver Knowledge Search (Naver KiN) [9], Baidu Zhidao [12], and Yahoo! Answer [10], as these websites are linearly growing. In these websites, the people answer the question that is send by other users, and these websites are specified as community-based question answering service (cQA) [4, 30], which are widely used in various fields [38]. However, the community cQAS offers the platform to allow the user to post the questions and answers that is already posted by some other user. In CQAS, the questions are openly available to receive the answers for a certain period of time. However, the asker can select the efficient and the best answer to the question along with the rating in the specified range. Moreover, each question has a list of an attribute called tag of interests that specifies the number of users who are interested in the question [7, 31].

QC plays a significant role in the question answering system and one of the important tasks of the classification process is to find the types of questions. However, most of the QAS tries to increase the technology and the experience of the user in finding the matched result. Due to the low response rate and the increasing amount of web content, finding the relevant content faces a challenging issue [2, 16, 32]. One of the methods used to perform the QC is to find the type of question with respect to the keywords and the structure of the sentence that specifies the semantic and the syntactic information, respectively. However, the set of patterns are specified with the regular expression often with hard-coded. When the user

enters a new question, it is matched with the set of patterns to find the class that belongs to. When the pattern gets more accurate and complete, the performance of the approach will be enhanced. In order to solve the performance issue, the questions are defined with numerous patterns [17, 18]. To achieve the classification strategy, different machine learning classifiers, such as Naive Bayes, K-Nearest Neighbor (KNN), auto-encoders [37] and support vector machine (SVM) are widely used. QC is a bloom taxonomy that contains well-defined keywords and categories such that each category contains well defined features. The classifier is trained by the training examples such that the classifiers are used to classify the questions [20, 33, 34].

In [19], decision tree, Sparse Network of Winnows (SNoW), naive bayes, and KNN are used to classify the English questions, where the bag-of-n gram features and the bag-of word features are effectively selected and showed that the SVM classifier is highly effective than the other learning classifiers. cQA services maintain and construct a large number of archives that include answering to the questions and the previous questions. As the cQA can directly answer to the queried questions rather than the list of documents such that the pairs of question-answer with large scale data is transformed into the information resource. If the relevant question and the answers are available, then the user can retrieve the answer from the list of relevant documents. Hence, question retrieval is a major factor in cQA service [4]. With the cQA services, retrieval of the question becomes a major part of the knowledge and information acquisition. The user expresses relevant meanings with the semantically related result that generates the lexical gap among similar questions [21]. Question retrieval is the major archive of cQA services that aims to retrieve the question answer pairs that are relevant to the query question. Other methods used to perform the question retrieval include the syntactic structure of questions and learning to rank methods [22].

This research focused to develop a question classification and retrieval approach based on the Bayesian probability and Tanimoto based RNN. The proposed Bayesian probability and Tanimoto based RNN is composed of three different engines, namely question relevancy prediction engine, question matching engine, and question retrieval engine. The proposed approach efficiently identifies the quality of answers that are relevant to the question. Here, the high quality of answers is accurately selected for the given questions. The temporal interactions between the answers are eliminated with the proposed approach. RNN classifier enables the question relevancy prediction engine to find the relevant questions based on the new loss function associated

with the RNN classifier. In the question matching engine, the dictionary based score is calculated using Planning Environment & Social Management (PESM), the position-based score is computed using Bayesian probability, and the common similarity is computed using the Tanimoto measure. With these three scores, the hybrid score is evaluated such that the question retrieval engine retrieves the relevant result based on the result of predicted value and the hybrid score value.

Contribution of the paper:

- The major contribution of this work is the development of a question classification and retrieval approach using the proposed Bayesian probability and Tanimoto based RNN classifier. The questions are classified and are retrieved through the question retrieval engine based on the predicted result of RNN and the hybrid score value of the question matching engine and reveal the effectiveness of user experience in retrieving the relevant result.
- The proposed Bayesian probability and Tanimoto based RNN classifier can solve the lexical gap issues and it can also estimate the relevancy between the questions.

The paper is organized as follows: Section 2 describes the review of the existing question classification and retrieval methods, and Section 3 elaborates the proposed Bayesian probability and Tanimoto for question classification and retrieval. Section 4 discussed the result and discussion of the proposed approach, and finally, Section 5 concludes the paper.

In this section, some of the existing question classification and retrieval methods are surveyed along with their merits and demerits, which motivate the researchers to develop a new question retrieval method in QAS.

2 Literature Survey

Various existing question retrieval techniques are reviewed in this section. Kyoungman Bae, and Youngjoong Ko [1] introduced a dependency based model (DM) and category-based model (CM) for increasing the performance of question retrieval. It effectively solved the issues of the lexical gap using the DM and CM models. However, the expansion words were selected using the expansion model. It attained better retrieval performance but failed to paraphrase the document using document and word embedding. Alaa Mohasseb et al. [2] developed a grammar-based approach for query classification and categorization. This method effectively exploited the structure

of questions and classified them. Here, the text was transformed into the pattern-based grammatical representation. It attained better performance but does not analyze and examine more questions from the various dataset. Partha Sarathy Banerjee et al. [3] developed a natural language information interpretation and representation (NLIIPS) model for question answering system. It eliminated the transformation of user query to the Structured Query Language (SQL) statements and avoided the storage of unstructured documents. It required less memory and no training process. It failed to consider the performance of the sentence with conjunctions. Kyoungman Bae, and Youngjoong Ko [4] developed a weighting method to achieve query classification. It effectively solved the issues of the lexical gap in the question retrieval system. It increased the performance of query retrieval and classification. However, it failed to consider the entity linking model to extract the features.

Chi-Hua Chen [5] developed an ensemble Neural Network (NN) for the question answering system. It used the training data with a random number in the training phase. It received the requirements of input from users and responded it with a suitable class of queries. This model obtained the highest accuracy, but the time complexity was too high. Fei Wu et al. [6] developed a temporal interaction and causal influence based long short term memory (TC-LSTM) for leveraging the casual and the temporal interaction between the question and answer. It effectively performed the answer ranking and answer classification tasks. It achieved better performance in answer selection but failed to use the attention model for discovering the semantic relations. Jinwei Liu et al. [7] developed coupled semi-supervised mutual reinforcement-based label propagation (CSMRLP) for quality prediction and analysis in the question answering system. It effectively generated the answers with high quality from the question features. It achieved better classification performance. It failed to analyze the social features to enhance the evaluation of question quality. Alami Hamza et al. [8] introduced a machine learning classifier in order to classify the Arabic questions. The representation of queries was calculated that allowed the user to capture the semantic and the syntactic relations between the words. It achieved better classification performance but failed to achieve the answer processing and document processing. Bei Xu and Hai Zhuge [35] have analyzed the influence of semantic links on the basic abilities of a type of QAS system that extracts answers from a range of texts. This analysis concerns the ability to answering different types of questions and supporting different patterns of answering questions. Asad Abdi et al. [36] have modelled a QAS in which the query and documents

Table 1 Features and challenges of existing question answering system

| Authors [Citations] | Methodology | Advantages | Disadvantages |
|--------------------------------------|---|---|---|
| Kyoungman Bae, and Youngjoong Ko [1] | DM and CM | <ul style="list-style-type: none"> • It is easy to solve the lexical gap • Better retrieval performance | <ul style="list-style-type: none"> • It is failed to paraphrase the document using document and word embedding. |
| Alaa Mohasseb et al. [2] | Grammar-based approach | <ul style="list-style-type: none"> • Easy to transform the text into the grammar-based approach | <ul style="list-style-type: none"> • It does not analyze and examine more questions from the various dataset |
| Partha Sarathy Banerjee et al. [3] | NLIIPS | <ul style="list-style-type: none"> • Avoids storage of unstructured document • Requires less memory and no training process | <ul style="list-style-type: none"> • Failed to consider the performance of the sentence with conjunctions |
| Kyoungman Bae, and Youngjoong Ko [4] | Case frame-based retrieval model with word embedding (WCFM) | <ul style="list-style-type: none"> • Easy to solve the issues of the lexical gap in the question retrieval system. • It can increase the performance of query retrieval and classification. | <ul style="list-style-type: none"> • It fails to consider the entity linking model to extract the features. |
| Chi-Hua Chen [5] | NN | <ul style="list-style-type: none"> • Easy to train the data with random numbers. • Highest accuracy | <ul style="list-style-type: none"> • Time complexity is very high |
| Fei Wu et al. [6] | TC-LSTM | <ul style="list-style-type: none"> • It can achieve better performance in answer selection | <ul style="list-style-type: none"> • Difficult to use the attention model for discovering the semantic relations |
| Jinwei Liu et al. [7] | CSMRLP | <ul style="list-style-type: none"> • Easy to generate the answers with high quality of question features. • It can achieve better classification performance. | <ul style="list-style-type: none"> • It is difficult to analyze the social features to enhance the evaluation of question quality. |

(Continued)

Table 1 Continued

| Authors [Citations] | Methodology | Advantages | Disadvantages |
|------------------------------|---|--|---|
| Alami Hamza et al. [8] | Machine-learning classifier | <ul style="list-style-type: none"> • Easy to capture the semantic and the syntactic relations between the words can achieve better classification performance | <ul style="list-style-type: none"> • Difficult to achieve the answer processing and document processing. |
| Asad Abdi et al. [36] | Question Answering System in Al-Hadith using Linguistic Knowledge (ASHLK) | <ul style="list-style-type: none"> • It is a time-saving process. • It can interpret natural language. | <ul style="list-style-type: none"> • It fails to capture the meaning of a sentence and a user's query. |

are processed by the question analyzer. It includes a set of basic linguistic functions, such as tagger and tokenizing. It can interpret natural language. It fails to capture the meaning when comparing a sentence and a user's query; hence there is often a conflict between the extracted sentences and user's requirements.

Due to the complexity issues, like the presence of diacritical marks, inflectional nature, absence of capital letters, and derivational related to complex morphology of the language, the question classification results a great challenge in the cQA system [8]. Question answer matching associated with the cQA poses a challenging issue, as it required efficient and effective representation, which captures the semantic relations between the question and answer [27]. The major challenge faced by the question retrieval mechanism is the feature sparsity, as the title of the question has a short length keyword with irregular noise. It is very complex to extract the modeling word from the entire question information [26]. The DM and the CM model in [1] effectively enhance the performance of retrieval, but the usage of external language resources, like wordnet, poses a challenging issue for the English words. To solve the issues of the lexical gap in order to estimate the relevance among the question poses a challenging issue in the question retrieval based cQA service.

To solve the lexical gap issues and to estimate the relevancy between the questions is a significant research issue in the QAS. Hence, a question classification and retrieval approach is developed in this research using bayesian probability and tanimoto based rnn classifier to achieve effective retrieval performance.

3 Proposed Hybrid Question Retrieval Approach Using Recurrent Neural Network

The proposed query classification and retrieval approach involves two different engines, namely question relevancy prediction engine and question matching engine. In the question relevancy prediction engine, the user enters the input question such that the words present in the input question is converted into the word sequential representation format, which is in the form of vector. The word sequential representation form is fed as the input to the classifier, where the question classification is carried out by the RNN classifier. In the question matching engine, three different scores, like dictionary-based score, position-based score, and common similarity is computed. However, the dictionary based score is calculated using PESM, position based score is computed using Bayesian probability, and common similarity is calculated using the Tanimoto measure. The hybrid score is evaluated based on the dictionary-based score, position-based score, and common similarity. Finally, the classified result obtained from the RNN classifier, and the hybrid score of question matching engine are fused together in the question retrieval engine. When the user enters the query, the query is matched with the question retrieval engine, and the relevant answer for the query is effectively retrieved. Figure 1 represents the schematic diagram of the proposed Bayesian probability and Tanimoto based RNN.

3.1 Question Relevancy Prediction Engine

In the question relevancy prediction engine, the user enters the input question to the question phase. The words present in the question are converted into the word sequential representation format, which is in the form of vector. Let us consider the user question as Q and the words present in the query Q are represented by their corresponding index in the word set W . Here, the word embedding matrix is designed to learn the word representation. The words in the query are converted into the word sequential representation format, which

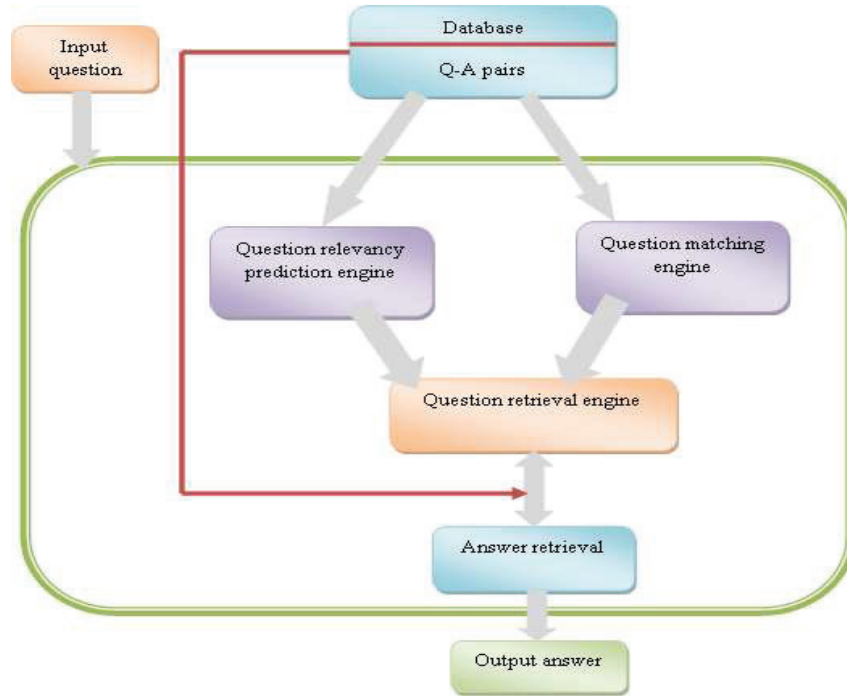


Figure 1 Schematic diagram of the proposed hybrid question retrieval approach using RNN.



Figure 2 Schematic diagram of question relevancy prediction engine.

is represented as F . Figure 2 shows the schematic diagram of the question relevancy prediction engine.

(i) Question relevancy prediction using RNN

The word sequential representation vector F is fed to the classification module, where the query relevancy prediction is carried out using the RNN classifier. RNN [28] takes the feature F as input and process the question

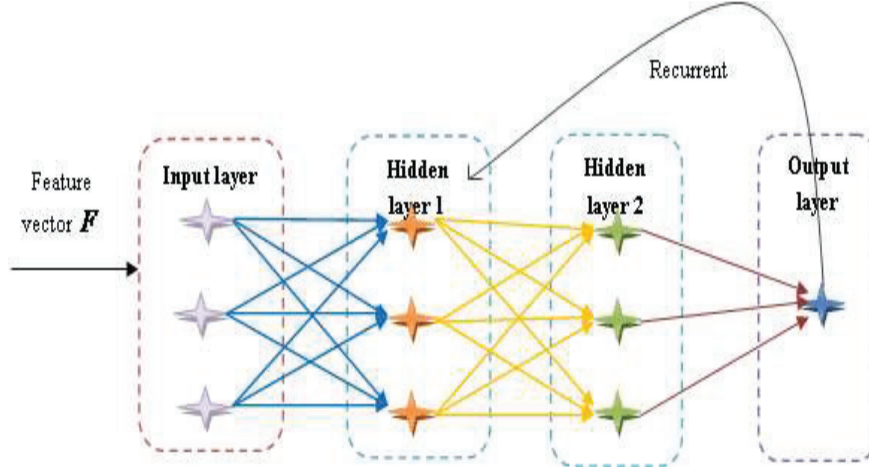


Figure 3 Architecture of RNN.

relevancy prediction based on the newly designed loss function using the Tanimoto measure. RNN is the type of supervised machine learning approach, which consists of artificial neurons with one recurrent loop termed as a feedback loop. The benefit of using RNN is to reduce the retrieval time between the query and the answer and to increase the retrieval result. Figure 3 represents the architecture of the RNN classifier.

It is capable to process the inputs based on the recurrent hidden state such that the activation function is based on the previous step. The standard RNN classifier is formulated as follows: Let us consider the sequence of inputs as $F_j (j = 1, 2, \dots, n)$, the sequence of hidden states as $H_j (j = 1, 2, \dots, n)$, and the sequence of predictions as $Y_p (p = 1, 2, \dots, n)$. However, the input is given to the hidden units and the output units of the RNN classifier is specified as, $a_j (j = 1, 2, \dots, n)$ and $g_j (j = 1, 2, \dots, n)$, respectively. The input given to the hidden layer is represented as,

$$a_j = W_{xy}F_j + W_{yy}F_{j-1} + B_v \quad (1)$$

where, a_j denotes the input of the hidden layer, W_{xy} represents the weight between the input and the hidden layer, W_{yy} denotes the weight associated between two hidden layers, and B_v represents the bias vector, respectively. However, the hidden state representation of RNN classifier is specified as,

$$H_j = \lambda(a_j) \quad (2)$$

where, a_j denotes the pre-defined vector values function. The output unit of RNN is represented using the below equation as,

$$g_j = W_{yz}H_j + B_s \quad (3)$$

Here, W_{yz} denotes the weight connected between the hidden unit and the output unit, and B_s indicates the bias vector. Finally, the predicted result is represented as,

$$Y_p = \beta(g_j) \quad (4)$$

where, β denotes the pre-defined vector values function. Finally, the result obtained by the RNN classifier is denoted as, S_m^{RNN} such that $S_m^{RNN} = Y_p$. The loss function associated with the RNN classifier is modified using the Tanimoto measure and is represented as,

$$X_1 = - \sum_{p=1}^k Y_o \log(Y_p) \quad (5)$$

$$X_2 = - \log \left(\sum_{p=1}^k \frac{(Y_o \cap Y_p)}{Y_o \cup Y_p} \right) \quad (6)$$

$$X_{loss} = \frac{X_1 + X_2}{2} \quad (7)$$

Here, X_{loss} is the loss function of RNN, which is obtained by modifying the loss function of RNN with the Tanimoto measure. Loss function is used to measure the deviation of predictions Y_p from the actual labels Y_o .

3.2 Question Matching Engine

The question matching engine contains three different scores, such as dictionary based score, position-based score, and common similarity. Here, the dictionary based score is computed using PESM, position-based score is calculated using Bayesian probability, and the common similarity is calculated using the Tanimoto measure. The hybrid score that is computed using the question matching engine is discussed as follows:

(i) Dictionary based score using PESM

The dictionary based score is calculated using PESM [23] based on the feature vector. PESM is the similarity degree used to retrieve the matched

result from the dictionary and calculate the dictionary based score. The dictionary contains numerous words, which is verified with the query entered by the user in order to compute the dictionary based score. Let us consider the user query as Q^{qry} , which have the question as “How are you?”. When the user enters a query to the question matching engine, the dictionary based score is computed by the question matching engine using PESM and returns the dictionary based score result. Let us assume the dictionary query as Q_i , which have the question as “What are you doing?”. The user query Q^{qry} is matched with the dictionary query Q_i and compute the dictionary based score by converting the user query into the dictionary based feature vector. From the feature vector, the dictionary based score is calculated using PESM. PESM is the measure used to compute the dictionary based score of the user query. However, the dictionary based score computed using PESM is represented as,

$$S_1(Q^{qry}, Q_i) = PESM(Q^{qry}, Q_i) \quad (8)$$

Here, S_1 represents the dictionary based score, and $PESM$ specifies the PESM measure, respectively.

(ii) Position based score using Bayesian probability

The second score to be calculated in the question matching engine is the position based score, which is calculated based on the Bayesian probability. The Bayesian probability is the interpretation of the concept of probability that represents the probability value for each word present in the query. However, the probability value for each word in the user query is calculated by matching the user query with the dictionary query. The user query consists of three words, such as ‘how’, ‘are’, and ‘you’. Each word present in the user query is matched with the words situated in the dictionary query to compute the probability value for each word. Let us specify the words ‘how’ as w_1 , ‘are’ as w_2 , and ‘you’ as w_3 , respectively. The position based score computed using the Bayesian probability is represented as,

$$S_2(Q^{qry}, Q_i) = P(w_1) * P(w_2) * P(w_3) * P(w_1, w_2) * P(w_2, w_3) \quad (9)$$

where, S_2 represents the position based score. Here, the probability of w_1 is represented as $P(w_1)$ and is calculated as $P(w_1) = 0/4$, as the word ‘how’ is not found in the dictionary query Q_i , the probability of w_2 is denoted as $P(w_2)$, which is calculated as $P(w_2) = 1/4$, as the word ‘are’ is found at one

time in the query Q_i , and the probability of a word w_3 is specified as $P(w_3)$, which have the values of $P(w_3) = 1/4$, as the word ‘you’ is found at one time in the query Q_i , respectively. The probability of combining two words, such as w_1 and w_2 is represented as $P(w_1, w_2)$, which have the values of $P(w_1, w_2) = 0/3$, as the word ‘how are’ is not found in the dictionary query Q_i , and the probability of combining the words w_2 and w_3 is represented as, $P(w_2, w_3)$, which have the values of $P(w_2, w_3) = 1/3$, as the word ‘are you’ is found at one time in the dictionary query Q_i .

(iii) Common similarity using Tanimoto

Common similarity is a score that is calculated using the Tanimoto measure [24] in the question matching engine. However, the common similarity score computed based on the Tanimoto measure is expressed as,

$$S_3(Q^{qry}, Q_i) = -\log_2 \left(\frac{Q^{qry} \cap Q_i}{Q^{qry} \cup Q_i} \right) \tag{10}$$

$$S_3(Q^{qry}, Q_i) = -\log_2 \left(\frac{2}{5} \right) \tag{11}$$

where, S_3 represents the common similarity value.

(iv) Hybrid score

The question matching engine calculates the hybrid score based on the dictionary based score, position based score, and common similarity. Therefore, the hybrid score computed in the question matching engine is represented as,

$$S_m^{match}(Q^{qry}, Q_i) = \frac{(S_1 + S_2 + S_3)}{3} \tag{12}$$

Here, S_m^{match} represents the hybrid score of question matching engine.

3.3 Question Retrieval Engine

In the question retrieval engine, the predicted result of the RNN classifier obtained using the question relevancy prediction engine and the hybrid score obtained from the question matching engine are fused together to retrieve the relevant questions. The question retrieval engine effectively retrieves the relevant question and returns the answers for the matched questions. Here, the questions retrieved based on the predicted result of RNN and the hybrid

score of matching engine is represented as,

$$Q_{01} = \underset{m=1}{\text{Max}}^N S_m^{RNN} \quad (13)$$

$$Q_{02} = \underset{m=1}{\text{Max}}^N S_m^{match} \quad (14)$$

Here, Q_{01} denotes the question retrieved based on the result of RNN, and Q_{02} represents the question that is retrieved based on the hybrid score value. Finally, the question selected for retrieving the answer is based on the following condition,

$$Q_{out} = \begin{cases} Q_{01} & \text{if } Q_{01} = Q_{02} \\ Q_{02} & \text{if } Q_{01} \neq Q_{02} \text{ and } S_m^{match} > S_m^{RNN} \\ Q_{01} & \text{if } Q_{01} \neq Q_{02} \text{ and } S_m^{match} \leq S_m^{RNN} \end{cases} \quad (15)$$

Here, Q_{out} represents the selected question used to retrieve the final answer using the question retrieval engine. Finally, the answer obtained for the selected query will be returned back to the user.

4 Results and Discussion

The results and discussion made using the proposed Bayesian probability and Tanimoto based RNN in terms of precision, recall, and F-measure are elaborated in this section.

4.1 Experimental Setup

The implementation of the proposed Bayesian probability and Tanimoto based RNN is carried out in the PYTHON tool using the question answer dataset specified in [25]. This dataset contains three different question files, namely S08, S09, and S10. Among the three question files, the question file S08, and the question file S09 are considered as dataset-1, and dataset-2, which are used to obtain the experimental result. The question answer file contains both the question and the answers. The parameters used for the experimentation is listed in Table 2.

4.2 Evaluation Metrics

The performance of the proposed method is evaluated using the metrics, like precision, recall, and F-measure.

Table 2 Parameters setup

| Parameters | Value |
|-----------------------------------|-------|
| Dropout rate | 0.1 |
| Number of epochs | 50 |
| Activation function | ReLU |
| Number of hidden layers | 2 |
| Neuron count in each hidden layer | 100 |

Precision: It is the measure used to specify the documents that are retrieved for the query, which is specified as,

$$R = \frac{K}{K + L} \quad (16)$$

Where, R indicates the precision measure, K denotes a true positive rate, and L denotes a false positive rate, respectively.

Recall: It is the measure used to retrieve the relevant documents based on the query, which is represented as

$$C = \frac{K}{K + V} \quad (17)$$

where, C represents recall, and V indicates a false-negative rate.

F-measure: A measure that combines both the precision and recall measures are specified as F-measure, which is represented as,

$$v = 2 \frac{R.C}{R + C} \quad (18)$$

Here, v indicates F-measure.

4.3 Comparative Methods

The performance enhancement of the proposed approach is revealed by comparing the proposed with the existing methods, like Dependency based model and Category based model (DM+CM) [1], Question Classification using a grammar-based approach (GQCC) [2], Augmented Reality question answering system (AR-QAS) [5], WCFM [4], and TC-LSTM [6], respectively.

4.4 Comparative Analysis

The comparative analysis of the proposed Bayesian probability and Tanimoto based RNN with respect to the performance metrics is discussed in this section.

4.4.1 Comparative analysis using dataset-1

The comparative analysis made using the dataset-1 based on the missing and the scrambled words is discussed in this section.

(i) Analysis based on missing words

The question with the missing words is selected as input and this question is processed using the proposed approach in order to evaluate the experimental result based on the missing words. Figure 4 portrays the comparative analysis of the proposed Bayesian probability and Tanimoto based RNN in terms of precision, recall, and F-measure for the missing words. Figure 4(a) represents the comparative analysis of precision by varying the number of questions. When the number of questions = 400, the precision obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 83.541%, 87.003%, 87.175%, 85.884%, and 87.321%, while the proposed Bayesian probability and Tanimoto based RNN obtained better precision of 87.444%, respectively. When the number of questions = 800, the precision obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 84.8%, 88.117%, 88.562%, 86.729%, and 88.643%, while the proposed Bayesian probability and Tanimoto based RNN obtained better precision of 89.055%, respectively.

Figure 4(b) represents the comparative analysis of recall by varying the number of questions. When the number of questions = 400, the recall obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 81.237%, 83.519%, 83.699%, 83.143%, and 83.804%, while the proposed Bayesian probability and Tanimoto based RNN obtained better recall of 84.162%, respectively. When the number of questions are increased to 800, the recall obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 81.983%, 83.588%, 84.113%, 83.563%, and 84.153%, while the proposed Bayesian probability and Tanimoto based RNN obtained better recall of 84.741%, respectively.

Figure 4(c) represents the comparative analysis of F-measure by varying the number of questions. When number of questions = 600, the F-measure obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 78.368%, 87.420%, 88.062%, 85.984%, and 88.345%, while the

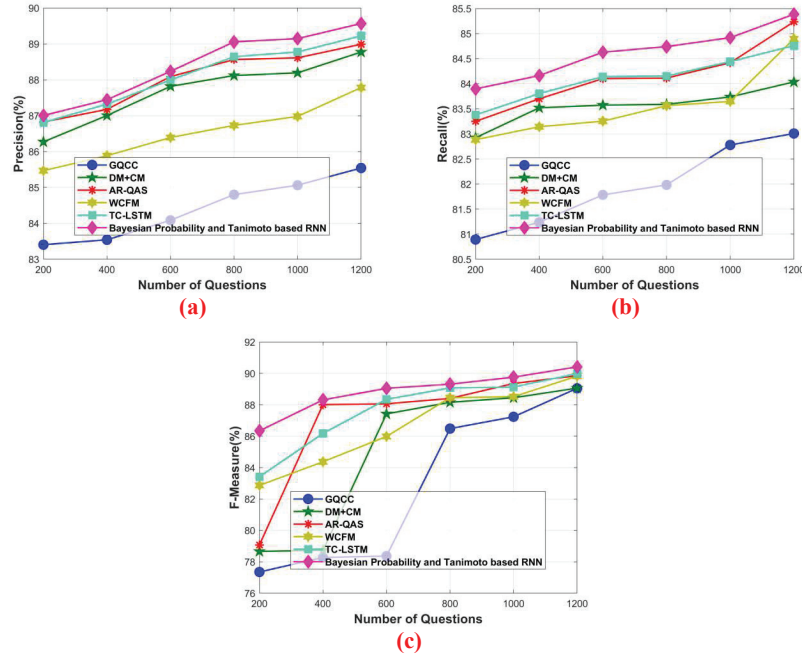


Figure 4 Comparative analysis based on missing words, (a) precision, (b) recall, (c) F-measure.

proposed Bayesian probability and Tanimoto based RNN obtained better F-measure of 89.050%, respectively. When the number of questions = 800, the F-measure obtained by the existing GQCC, DM + CM, AR-QAS, WCFM, and TC-LSTM is 86.483%, 88.158%, 88.405%, 88.45%, and 89.074%, while the proposed Bayesian probability and Tanimoto based RNN obtained better F-measure of 89.312%, respectively. It is shown that the proposed Bayesian probability and Tanimoto based RNN obtained stable result for 1200 questions such that the performance gets degrade by increasing the number of questions.

(ii) Analysis based on scrambled words

The words in the questions are scrambled somewhere and the question with the scrambled words are given as the input to the classifier to generate the experimental result based on scrambled words. Figure 5 portrays the comparative analysis of the proposed Bayesian probability and Tanimoto based RNN in terms of precision, recall, and F-measure for the scrambled words. Figure 5(a) represents the comparative analysis of precision by varying the

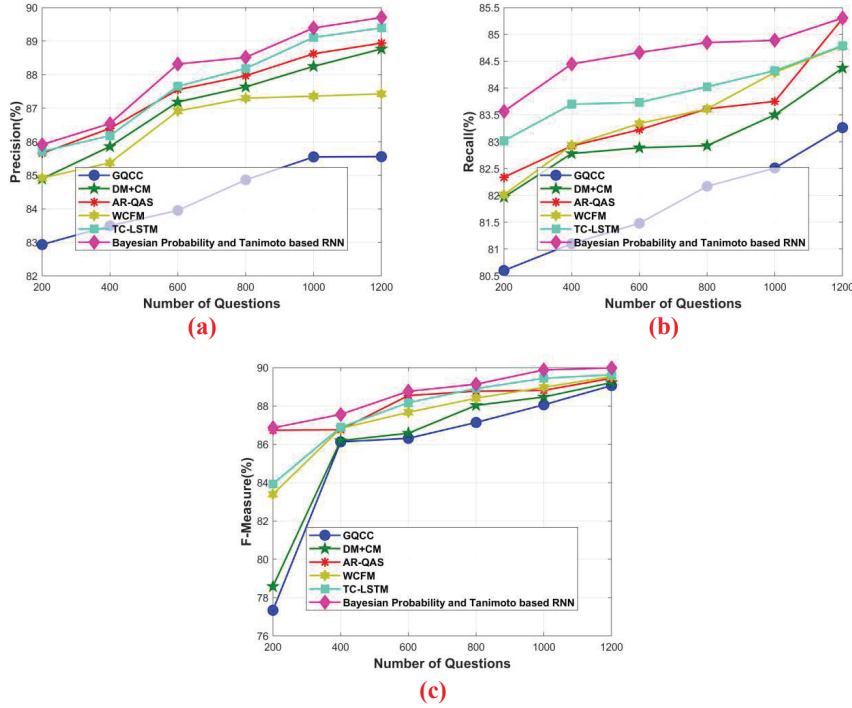


Figure 5 Comparative analysis based on scrambled words, (a) precision, (b) recall, (c) F-measure.

number of questions. When the number of questions = 400, the precision obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 83.496%, 85.858%, 86.396%, 85.371%, and 86.178%, while the proposed Bayesian probability and Tanimoto based RNN obtained better precision of 86.541%, respectively. When the number of questions = 800, the precision obtained by the existing GQCC, DM + CM, AR-QAS, WCFM, and TC-LSTM is 84.863%, 87.635%, 87.966%, 87.298%, and 88.185%, while the proposed Bayesian probability and Tanimoto based RNN obtained better precision of 88.515%, respectively.

Figure 5(b) represents the comparative analysis of recall by varying the number of questions. When number of questions = 400, the recall obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 81.099%, 82.780%, 82.920%, 82.934%, and 83.699%, while the proposed Bayesian probability and Tanimoto based RNN obtained better recall of 84.448%, respectively. When number of questions = 800, the recall obtained

by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 82.170%, 82.926%, 83.612%, 83.61%, and 84.024%, while the proposed Bayesian probability and Tanimoto based RNN obtained better recall of 84.847%, respectively.

Figure 5(c) represents the comparative analysis of F-measure by varying the number of questions. When the number of questions = 400, the F-measure obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 86.123%, 86.186%, 86.754%, 86.825%, and 86.879%, while the proposed Bayesian probability and Tanimoto based RNN obtained better F-measure of 87.55%, respectively. When number of questions = 800, the F-measure obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 87.134%, 88.027%, 88.766%, 88.399%, and 88.903%, while the proposed Bayesian probability and Tanimoto based RNN obtained better F-measure of 89.1307%, respectively. From the below graph, it is clearly shown that the performance of the proposed Bayesian probability and Tanimoto based RNN is stable for around 1000 and 1200 questions. When the number of questions are increased to above 1200, the performance of the proposed Bayesian probability and Tanimoto based RNN obtained may get reduce.

4.4.2 Comparative analysis using dataset-2

The comparative analysis made using the dataset-2 based on the missing and the scrambled words is discussed in this section.

(i) Analysis based on missing words

The question with the missing words is selected as input and this question is processed using the proposed approach to evaluate the experimental result based on the missing words. Figure 6 portrays the comparative analysis of the proposed Bayesian probability and Tanimoto based RNN in terms of precision, recall, and F-measure with respect to the missing words. Figure 6(a) represents the comparative analysis of precision by varying the number of questions. When number of questions = 600, the precision obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 83.414%, 87.237%, 87.543%, 86.83%, and 87.612%, while the proposed Bayesian probability and Tanimoto based RNN obtained better precision of 87.894%, respectively. When number of questions = 800, the precision obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 84.579%, 88.298%, 88.350%, 87.435%, and 88.339%, while the proposed Bayesian probability and Tanimoto based RNN obtained better precision of 88.376%,

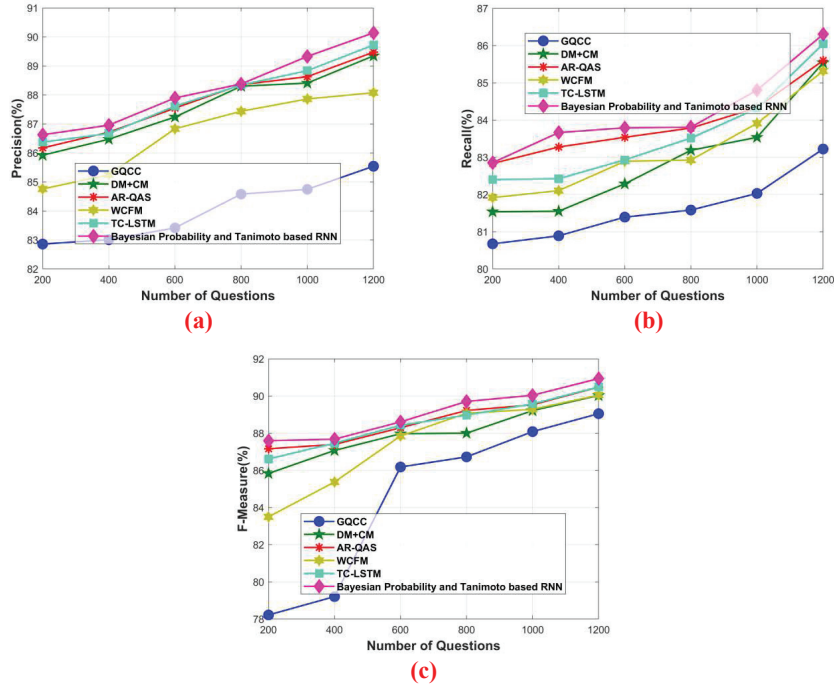


Figure 6 Comparative analysis based on missing words, (a) precision, (b) recall, (c) F-measure.

respectively. When number of questions = 1000, the precision obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 84.743%, 88.407%, 88.627%, 87.859%, and 88.834%, while the proposed Bayesian probability and Tanimoto based RNN obtained better precision of 89.325%, respectively.

Figure 6(b) represents the comparative analysis of recall by varying the number of questions. When number of questions = 400, the recall obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 80.889%, 81.549%, 83.274%, 82.103%, and 82.423%, while the proposed Bayesian probability and Tanimoto based RNN obtained better recall of 83.661%, respectively. When number of questions = 600, the recall obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 81.391%, 82.284%, 83.534%, 82.89%, and 82.928%, while the proposed Bayesian probability and Tanimoto based RNN obtained better recall of 83.789%, respectively. When number of questions = 1000, the recall obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and

TC-LSTM is 82.027%, 83.533%, 84.330%, 83.908%, and 84.309%, while the proposed Bayesian probability and Tanimoto based RNN obtained better recall of 84.799%, respectively.

Figure 6(c) portrays the comparative analysis of F-measure by varying the number of questions. When number of questions = 400, the F-measure obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 79.205%, 87.068%, 87.391%, 85.372%, and 87.459%, while the proposed Bayesian probability and Tanimoto based RNN obtained better F-measure of 87.676%, respectively. When number of questions = 800, the F-measure obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 86.723%, 87.998%, 89.219%, 89.072%, and 88.971%, while the proposed Bayesian probability and Tanimoto based RNN obtained better F-measure of 89.706%, respectively.

(ii) Analysis based on scrambled words

The words in the questions are scrambled somewhere and the question with the scrambled words are given as the input to the classifier to generate the experimental result based on scrambled words. Figure 7 portrays the comparative analysis of the proposed Bayesian probability and Tanimoto based RNN in terms of precision, recall, and F-measure with respect to the scrambled words. Figure 7(a) represents the comparative analysis of precision by varying the number of questions. When number of questions = 400, the precision obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 83.66%, 86.166%, 86.886%, 85.459%, and 86.943%, while the proposed Bayesian probability and Tanimoto based RNN obtained better precision of 87.165%, respectively. When number of questions = 1000, the precision obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 84.592%, 88.459%, 88.692%, 87.306%, and 89.028%, while the proposed Bayesian probability and Tanimoto based RNN obtained better precision of 89.236%, respectively.

Figure 7(b) represents the comparative analysis of recall by varying the number of questions. When number of questions = 400, the recall obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 81.379%, 82.230%, 83.792%, 83.046%, and 83.457%, while the proposed Bayesian probability and Tanimoto based RNN obtained better recall of 84.099%, respectively. When number of questions = 800, the recall obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 81.964%, 83.464%, 84.437%, 83.355%, and 84.199%, while the proposed Bayesian probability and Tanimoto based RNN obtained better

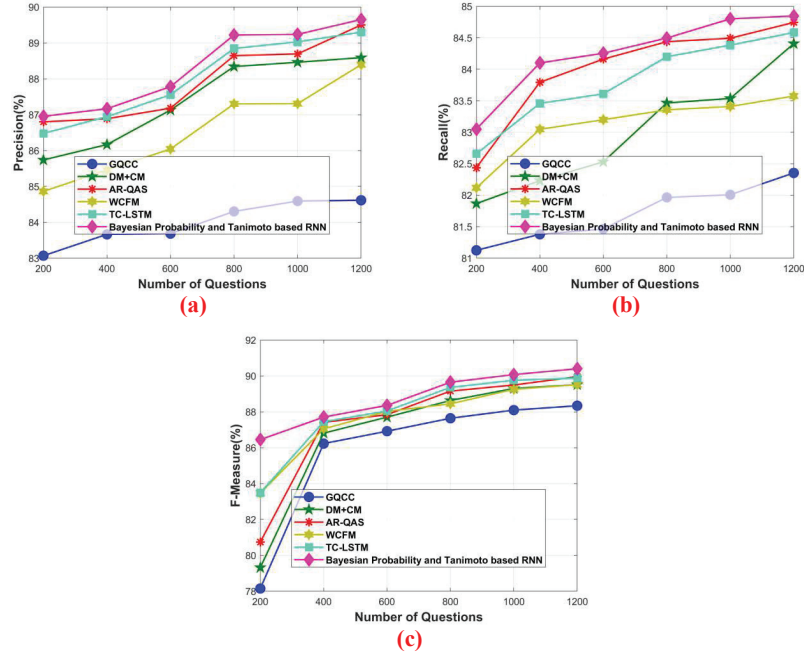


Figure 7 Comparative analysis based on scrambled words, (a) precision, (b) recall, (c) F-measure.

recall of 84.495%, respectively. Figure 7(c) shows the comparative analysis of F-measure by varying the number of questions. When number of questions = 400, the F-measure obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 86.227%, 86.807%, 87.411%, 87.046%, and 87.442%, while the proposed Bayesian probability and Tanimoto based RNN obtained better F-measure of 87.702%, respectively. When number of questions = 1000, the F-measure obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 88.092%, 89.299%, 89.490%, 89.251%, and 89.755%, while the proposed Bayesian probability and Tanimoto based RNN obtained better F-measure of 90.072%, respectively. From the below graph it is clearly stated that the performance get decrease by increasing the number of questions, as it stated a stable result for 1200 questions.

4.5 Comparative Discussion

Table 3 portrays the comparative discussion of the proposed Bayesian probability and Tanimoto based RNN. By considering the missing words, the

Table 3 Comparative discussion

| Metrics/Methods | | GQCC | DM+CM | AR-QAS | WCFM | TC-LSTM | Proposed Bayesian Probability and Tanimoto Based RNN |
|------------------------|---------------|---------------|--------|--------|--------|---------|--|
| Dataset-1 | Missing words | Precision (%) | 88.772 | 88.989 | 87.782 | 89.223 | 89.567 |
| | | Recall (%) | 84.037 | 85.236 | 84.895 | 84.759 | 85.386 |
| | | F-measure (%) | 89.049 | 89.062 | 89.856 | 89.827 | 90.415 |
| Scrambled words | | Precision (%) | 88.766 | 88.941 | 87.428 | 89.392 | 89.704 |
| | | Recall (%) | 84.372 | 85.294 | 84.779 | 84.79 | 85.305 |
| | | F-measure (%) | 89.047 | 90.475 | 90.049 | 90.487 | 90.936 |
| Dataset-2 | Missing words | Precision (%) | 89.345 | 89.473 | 88.075 | 89.719 | 90.14 |
| | | Recall (%) | 85.536 | 85.596 | 85.315 | 86.042 | 86.301 |
| | | F-measure (%) | 89.047 | 90.475 | 90.049 | 90.487 | 90.936 |
| Scrambled words | | Precision (%) | 88.589 | 89.498 | 88.392 | 89.301 | 89.65 |
| | | Recall (%) | 84.407 | 84.743 | 83.574 | 84.584 | 84.846 |
| | | F-measure (%) | 88.337 | 89.955 | 89.509 | 89.878 | 90.395 |

Table 4 Statistical analysis using Dataset-1

| | | Missing Words | | | Scrambled Words | | |
|---|----------|---------------|------------|---------------|-----------------|------------|---------------|
| | | Precision (%) | Recall (%) | F-measure (%) | Precision (%) | Recall (%) | F-measure (%) |
| GQCC | Mean | 85.537 | 83.004 | 89.044 | 85.551 | 83.259 | 89.041 |
| | Variance | 0.005 | 0.004 | 0.005 | 0.005 | 0.004 | 0.006 |
| DM+CM | Mean | 88.766 | 84.031 | 89.057 | 88.761 | 84.366 | 90.021 |
| | Variance | 0.006 | 0.006 | 0.005 | 0.005 | 0.006 | 0.005 |
| AR-QAS | Mean | 88.982 | 85.231 | 89.851 | 88.935 | 85.289 | 90.471 |
| | Variance | 0.007 | 0.005 | 0.005 | 0.006 | 0.005 | 0.004 |
| WCFM | Mean | 87.775 | 84.889 | 89.821 | 87.423 | 84.772 | 90.042 |
| | Variance | 0.007 | 0.006 | 0.006 | 0.005 | 0.007 | 0.007 |
| TC-LSTM | Mean | 89.218 | 84.755 | 89.977 | 89.387 | 84.785 | 90.482 |
| | Variance | 0.005 | 0.004 | 0.006 | 0.005 | 0.005 | 0.005 |
| Proposed Bayesian probability and Tanimoto based RNN | Mean | 89.564 | 85.384 | 90.413 | 89.701 | 85.303 | 90.935 |
| | Variance | 0.003 | 0.002 | 0.002 | 0.003 | 0.002 | 0.001 |

Table 5 Statistical analysis using Dataset-2

| | Missing Words | | | Scrambled Words | | |
|---|---------------|------------|---------------|-----------------|------------|---------------|
| | Precision (%) | Recall (%) | F-measure (%) | Precision (%) | Recall (%) | F-measure (%) |
| GQCC | Mean | 85.532 | 83.215 | 89.042 | 84.609 | 82.347 |
| | Variance | 0.007 | 0.006 | 0.005 | 0.006 | 0.006 |
| DM+CM | Mean | 89.339 | 85.531 | 90.020 | 88.581 | 84.401 |
| | Variance | 0.006 | 0.005 | 0.006 | 0.008 | 0.006 |
| AR-QAS | Mean | 89.467 | 85.591 | 90.469 | 89.491 | 84.736 |
| | Variance | 0.006 | 0.005 | 0.006 | 0.007 | 0.007 |
| WCFM | Mean | 88.069 | 85.308 | 90.041 | 88.385 | 83.568 |
| | Variance | 0.006 | 0.007 | 0.008 | 0.007 | 0.006 |
| TC-LSTM | Mean | 89.713 | 86.035 | 90.481 | 89.295 | 84.579 |
| | Variance | 0.006 | 0.007 | 0.006 | 0.006 | 0.005 |
| Proposed Bayesian probability and Tanimoto based RNN | Mean | 90.137 | 86.299 | 90.934 | 89.649 | 84.845 |
| | Variance | 0.003 | 0.002 | 0.002 | 0.001 | 0.001 |

precision obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 85.541%, 88.771%, and 88.989%, while the proposed Bayesian probability and Tanimoto based RNN obtained better precision of 89.567%, respectively using dataset-1. By considering missing words using dataset-1, the recall obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 83.008%, 84.036%, and 85.236%, while the proposed Bayesian probability and Tanimoto based RNN is 85.386%, respectively. By considering scrambled words using dataset-1, the precision obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 85.555%, 88.766%, and 88.940%, while the proposed Bayesian probability and Tanimoto based RNN is 89.704%, respectively. By considering missing words using dataset-2, the recall obtained by the existing GQCC, DM+CM, and AR-QAS is 83.220%, 85.536%, and 85.596%, while the proposed Bayesian probability and Tanimoto based RNN is 86.301%, respectively. By considering missing words using dataset-2, the F-measure obtained by the existing GQCC, DM+CM, AR-QAS, WCFM, and TC-LSTM is 89.047%, 90.026%, and 90.474%, while the proposed Bayesian probability and Tanimoto based RNN is 90.936%, respectively. Thus, it is shown that the proposed Bayesian probability and Tanimoto based RNN obtained better results using dataset-2 in terms of precision, recall, and F-measure with respect to the missing words.

4.6 Statistical Analysis

Table 4 illustrates the statistical analysis of dataset-1 for missing words and scrambled words with precision, recall and F-measure metrics.

Table 5 illustrates the statistical analysis of dataset-2 for missing words and scrambled words with precision, recall and F-measure metrics, and the mean and variance of the existing GQCC, DM+CM, AR-QAS, WCFM, TC-LSTM, and Proposed Bayesian probability and Tanimoto based RNN are determined.

5 Conclusion

In this research, a hybrid question classification and retrieval approach is developed using the proposed Bayesian probability and Tanimoto based RNN classifier. The proposed QAS consists of three engines, namely question relevancy prediction engine, question matching engine, and question retrieval engine. Initially, the question pairs are sent to the question relevancy prediction engine, where the relevant queries are retrieved, and question matching

engine, where the queries that are matched to the user is retrieved. In the question relevancy prediction engine, the user query is converted into the word sequential representation format. In the question matching engine, the hybrid score is calculated for each query to retrieve the relevant search result. Finally, the classified result obtained from the RNN classifier, and the hybrid score of question matching engine are fused together in the question retrieval engine. When the user enters the query, the query is matched with the question retrieval engine, and the relevant answer for the query is effectively retrieved. The performance of the proposed method is analyzed using the Question answer dataset and the results are compared with the existing methods, such as DM+CM, GQCC, and AR-QAS, WCFM, and TC-LSTM. From the results, it is shown that the proposed method obtained better performance in terms of precision, recall, and F-measure with the values of 90.14, 86.301, and 90.936 using dataset-2.

References

- [1] Kyoungman Bae, and Youngjoong Ko, “Improving Question Retrieval in Community Question Answering Service Using Dependency Relations and Question Classification”, *Journal of the association for information science and technology*, pp. 1–16, 2019.
- [2] Alaa Mohasseb, Mohamed Bader-El-Den, and Mihaela Cocea, “Question categorization and classification using grammar based approach”, *Information Processing and Management*, vol. 54, no. 6, pp. 1228–1243, 2018.
- [3] Partha Sarathy Banerjee, Baisakhi Chakraborty, Deepak Tripathi, Hardik Gupta, and Sourabh S. Kumar, “A Information Retrieval Based on Question and Answering and NER for Unstructured Information Without Using SQL”, *Wireless Personal Communications*, pp. 1–23, 2019.
- [4] Kyoungman Bae, and Youngjoong Ko, “Efficient question classification and retrieval using category information and word embedding on cQA services”, *Journal of Intelligent Information Systems*, pp. 1–23, 2019.
- [5] Chi-Hua Chen, Chen-Ling Wu, Chi-Chun Lo, and Feng-Jang Hwang, “An Augmented Reality Question Answering System Based on Ensemble Neural Networks”, *IEEE Access*, vol. 5, pp. 17425–17435, August 2017.
- [6] Fei Wu, Xinyu Duan, Jun Xiao, Zhou Zhao, Siliang Tang, Yin Zhang, and Yueting Zhuang, “Temporal Interaction and Causal Influence in

- Community-based Question Answering”, *Journal of latex class files*, vol. 14, no. 8, August 2015.
- [7] Jinwei Liu, Haiying Shen, Member, ACM, and Lei Yu, “Question Quality Analysis and Prediction in Community Question Answering Services with Coupled Mutual Reinforcement”, *IEEE Transactions on Services Computing*, vol. 10, no. 2, pp. 286–301, March 2017.
- [8] Alami Hamza, Noureddine En-Nahnahi, Khalid Alaoui Zidani, Said El Alaoui Ouatik, “An arabic question classification method based on new taxonomy and continuous distributed representation of words”, *Journal of King Saud University-Computer and Information Sciences*, 2019.
- [9] Kyoungman Bae and Youngjoong Ko, “An Effective Question Expanding Method for Question Classification in cQA Services”, November 2014.
- [10] Delphine Bernhard and Iryna Gurevych, “Combining Lexical Semantic Resources with Question & Answer Archives for Translation-Based Answer Finding”, In proceedings of 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pp. 728–736, 2009.
- [11] Adam Berger, and John Laerty, “Information Retrieval as Statistical Translation”, *SIGIR’99*, pp. 222–229, 1999.
- [12] Peter E Brown, and Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer, “The Mathematics of Statistical Machine Translation: Parameter Estimation”, *Association for Computational Linguistics*, 1993.
- [13] Haiying Shen, Ze Li, Jinwei Liu, Joseph Edward Grant, “Knowledge sharing in the online social network of yahoo! answers and its implications”, *IEEE Transactions on Computers*, vol. 64, no. 6, pp. 1715–1728, June 2015.
- [14] Anna Shtok, Gideon Dror, Yoelle Maarek, “Learning from the Past: Answering New Questions with Past Answers”, In proceedings of International World Wide Web Conference Committee, April 2012.
- [15] Lotfi A. Zadeh, “From Search Engines to Question Answering Systems – The Problems of World Knowledge, Relevance, Deduction and Precision”, In *Fuzzy Logic and the Semantic Web*, pp. 163–210, 2006.
- [16] Zhe Liu, Bernard J. Jansen, “Identifying and predicting the desire to help in social question and answering”, *Information Processing and Management*, pp. 1–15, 2016.
- [17] Wei Li, “Question Classification Using Language Modeling”, *CIIR technical report*, 2002.

- [18] Li Liu, Zhengtao Yu, Jianyi Guo, Cunli Mao, and Xudong Hong, “Chinese Question Classification Based on Question PropertyKernel”, *International Journal of Machine Learning and Cybernetics*, vol. 5, no. 5, pp. 713–720, 2014.
- [19] Dell Zhang, and Wee Sun Lee, “Question Classification using Support Vector Machines”, In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 26–32, 2003.
- [20] Anbuselvan Sangodiah, Rohiza Ahmad, and Wan Fatimah Wan Ahmad, “A Review in Feature Extraction Approach in Question Classification Using Support Vector Machine”, In *proceedings of IEEE International Conference on Control System, Computing and Engineering*, pp. 28–30 November 2014.
- [21] Kai Zhang, Wei Wu, Fang Wang, Ming Zhou, and Zhoujun Li, “Learning Distributed Representations of Data in Community Question Answering for Question Retrieval”, In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 533–542, 2016.
- [22] Xin Cao, Gao Cong, Bin Cui, and Christian S. Jensen, “A Generalized Framework of Exploring Category Information for Question Retrieval in Community Question Answer Archives”, In *Proceedings of the 19th international conference on World wide web*, pp. 201–210, 2010.
- [23] Poonam Yadav, “Case Retrieval Algorithm Using Similarity Measure and Adaptive Fractional Brain Storm Optimization for Health Informaticians”, *Arabian Journal for Science and Engineering*, vol. 41, no. 3, pp. 829–840, 2016.
- [24] Szabolcs Serghyan, “Color Histogram Features Based Image Classification in Content-Based Image Retrieval Systems”, In *proceedings of 6th International Symposium on Applied Machine Intelligence and Informatics*, pp. 221–224, 2008.
- [25] Question answer dataset taken from, “<https://www.kaggle.com/ratman/questionanswer-dataset/version/1>”, accessed on August 2019.
- [26] Zheqian Chen, Chi Zhang, Zhou Zhao, and Deng Cai, “Question Retrieval for Community-based Question Answering via Heterogeneous Network Integration Learning”, 2016.
- [27] Lorena Kodra, and Elinda Kajo Mece, “Question Answering Systems: A Review on Present Developments, Challenges, and Trends”, *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 9, 2017.

- [28] Martens J, Sutskever I, “Learning recurrent neural networks with hessian-free optimization”, In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 1033–1040, 2011.
- [29] Esposito, M., Damiano, E., Minutolo, A., De Pietro, G. and Fujita, H., “Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering”, *Information Sciences*, vol. 514, pp. 88–105, 2020.
- [30] Sarrouiti, M. and El Alaoui, S.O., “SemBioNLQA: A semantic biomedical question answering system for retrieving exact and ideal answers to natural language questions”, *Artificial Intelligence in Medicine*, vol. 102, p. 101767, 2020.
- [31] Yilmaz, T., Ozcan, R., Altıngövdü, I.S. and Ulusoy, Ö., “Improving educational web search for question-like queries through subject classification”, *Information Processing & Management*, vol. 56, no. 1, pp. 228–246, 2019.
- [32] Yang, S. and Gao, C., “Enriching basic features via multilayer bag-of-words binding for Chinese question classification”, *CAAI Transactions on Intelligence Technology*, vol. 2, no. 3, pp. 133–140, 2019.
- [33] Kundu, D. and Mandal, D.P., “Formulation of a hybrid expertise retrieval system in community question answering services”, *Applied Intelligence*, vol. 49, no. 2, pp. 463–477, 2019.
- [34] Madaan, R., Sharma, A.K., Dixit, A., and Bhatia, P., “Indexing of Semantic Web for Efficient Question Answering System”, In *Software Engineering*, Springer, Singapore, pp. 51–61, 2019.
- [35] Bei Xu, and Hai Zhuge, “The influence of semantic link network on the ability of question-answering system”, *Future Generation Computer Systems*, vol. 108, pp. 1–14, 2020.
- [36] Asad Abdi, Shafaatunnur Hasan, Mohammad Arshi, Siti Mariyam Shamsuddin, and Norisma Idris, “A question answering system in hadith using linguistic knowledge”, *Computer Speech & Language*, vol. 60, 2020.
- [37] Vaibhav Rupapara, Manideep Narra, Naresh Kumar Gonda, Kaushika Thipparthi, and Swapnil Gandhi, “Auto-Encoders for Content-based Image Retrieval with its Implementation Using Handwritten Dataset”, In the proceeding of 5th International Conference on Communication and Electronics Systems (ICCES), IEEE, 289–294, 2020.

- [38] H Sumit, Sailee Bhambere and Abhishek B, “Rapid Digitization of Healthcare – A Review of COVID-19 Impact on our Health systems”, *International Journal of All Research Education and Scientific Methods*, vol. 9, no. 2, pp. 1457–1459, 2021.

Biography



Veeraraghavan Jagannathan is a researcher in Machine Learning, NLP and deep learning. He received a Post Graduate in Computer Science and Engineering from Anna University, Chennai, India and obtained his Ph.D Degree in Computer Science, from the prestigious National Institute of Technology, Trichy, India in 2008 and 2017 respectively. He has over a decade of research experience and 20 years of academic experience. He has published papers in reputed international journals. His current research areas include, but not limited to Data Analytics, GANs for NLP, Computer Vision and CNN, Medical Image Processing, and Medical Data Analytics and predictive analytics.

