International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Maximizing Answer Relevance in Community Question Answering with Low Computational Overhead: Insights from Clustering and Fuzzy-Ranking

Manisha Vilas Khadse[1], Dr. Neeraj Sahu[2]

Submitted: 06/12/2023 **Revised**: 12/01/2024 **Accepted**: 30/01/2024

Abstract: The primary objective of a question answering system is to retrieve high-quality answers, yet numerous new questions often remain unanswered effectively. In response to this challenge, we present a novel approach that enables users to input a question, receiving a collection of closely related questions. This method aims to provide users with satisfactory responses promptly, eliminating the need for extended waits for answers from other users. Our approach employs a string similarity coupled with a clustering in order to retrieve and group questions that are similar or closely related from the dataset. However, addressing the problem isn't merely about ranking similar question-answer pairs. We also take into account expert ratings of answers, treating them as indices or ratings of user satisfaction. To incorporate these ratings into our framework, we utilize Triangular Fuzzy Numbers (TFN). The average of TFN and the similarity index of the question yields a precise measure of answer satisfaction. In our experimental setup, we utilize a health domain dataset containing user-generated health-related questions. The experimental results conducted on health Community Question Answering (CQA) datasets affirm the superior performance of our proposed method. In predicting suitable answer for new questions, our method demonstrates a higher accuracy compared to other approaches. The effectiveness of our approach is particularly notable in the context of health-related inquiries, showcasing its potential to deliver more precise and reliable outcomes.

Keywords: *Community Question Answering (CQA), Triangular Fuzzy Numbers (TFN), String Similarity, Clustering, Similarity Index*

1. Introduction

Retrieving pertinent information and filtering out unnecessary data from extensive datasets are crucial for meeting user demands efficiently. Information retrieval methods commonly adopt a term-based approach due to its simplicity and efficient computational performance. The surge in popularity of question-and-answer services on public platforms, such as Yahoo Answers and Quora, has led to a substantial increase in user traffic. These platforms offer users the flexibility to submit queries, with experts worldwide contributing answers [1,5].

Efficient information retrieval is not only about obtaining correct answers but also ensuring timely responses for users who pose questions. To minimize wait times, it is imperative to locate similar questions and answers within historical document corpora. The traditional question-answering (QA) Pipeline architecture, depicted in Fig. 1, operates on domain-specific knowledge datasets. In this framework, users post questions, and the QA system searches for comparable or related questions already present in the dataset. The predominant method for document retrieval on the web involves keyword-based searches. However, the answer selection process encounters two challenging parameters. Firstly, numerous Community Question Answering (cQA) datasets are open-domain, encompassing questions that demand descriptive answers. In response to these challenges, we have devised a content-driven approach to answer selection. We have showcased the effectiveness of this approach in efficiently ranking lists of questions or answers related to a given query [6,14].

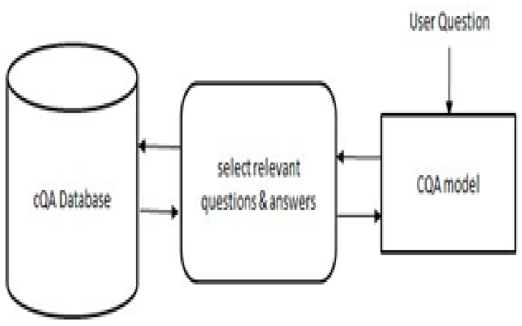

Fig. 1. A traditional QA pipeline architecture

1.1 Issues in Current CQA Portals

In the realm of question answering systems, the primary objective revolves around the retrieval of high-quality answers, yet the challenge persists where numerous new questions often go unanswered effectively.

[1] *G H Raisoni University, Amravati, Maharashtra, India*
ORCID ID : 0000-0002-3614-8184
[2] *G H Raisoni University, Amravati, Maharashtra, India*
ORCID ID : 0000-0001-9365-3761
** Corresponding Author Email: manishakhadse151@gmail.com*

Community Question Answering (CQA) models encounter various challenges that can impact their performance and user satisfaction. Some of the key issues in CQA models include:

1.1.1. Question Understanding

Ambiguity: Some questions may be poorly formulated or ambiguous, making it challenging for models to understand and provide accurate answers.

Contextual Understanding: Models may struggle to grasp the context of certain questions, impacting the relevance of responses.

1.1.2. Answer Redundancy and Overlap

Repetitive Answers: CQA models may produce similar or identical answers to a given question, leading to redundancy.

Overlapping Content: Answers may overlap in content, causing confusion for users.

1.1.3. Temporal Sensitivity

Outdated Information: CQA models may not effectively handle changes over time, leading to the dissemination of outdated information.

Dynamic Topics: Some domains or topics may evolve rapidly, requiring models to adapt to current trends.

1.1.4. User Engagement and Participation

Bias towards Popular Questions: Models may prioritize popular questions, neglecting less popular but equally important queries.

Limited User Participation: Low user engagement or inactive users can impact the model's ability to provide timely and diverse answers.

1.1.5. Trust and Credibility

Bias in Training Data: Models trained on biased data may perpetuate biases and impact the trustworthiness of responses.

Difficulty in Evaluating Credibility: Assessing the credibility of answers can be challenging, especially in open community settings.

1.1.6. Scalability and Efficiency

Scalability Issues: As the user base grows, models may face scalability challenges in handling increased volumes of data and user interactions.

Latency Concerns: Long response times may discourage user engagement, particularly for real-time or time-sensitive queries.

1.2. Computational Cost of Existing CQA Models

The computational cost of Community Question Answering (CQA) models can vary depending on the specific architecture, model complexity, and the size of the dataset being used. Here are some factors influencing the computational cost:

1.2.1. Model Architecture

Size and Complexity**:** Larger and more complex models, such as deep neural networks with numerous parameters, generally require more computational resources.

Attention Mechanisms: Models using attention mechanisms or transformer architectures may have higher computational costs due to increased parallelization.

1.2.2. Training and Fine-tuning

Training Data Size**:** The size of the training `dataset significantly affects computational costs. Larger datasets may require more processing power and time.

Fine-tuning: Some models, especially pre-trained ones, may require additional computational resources for fine-tuning on specific CQA datasets.

1.2.3. Embedding and Vectorization

Word Embeddings: Utilizing pre-trained word embeddings or contextual embedding’s can add to computational costs during model training and inference.

Vectorization Techniques: Techniques such as embedding lookup and vectorization contribute to the computational load, especially with large vocabularies [10,11,12].

1.2.4. Inference and Prediction

Batch Size: The batch size used during inference can impact computational costs. Larger batch sizes may lead to more efficient GPU utilization.

Real-time Processing: If real-time or near real-time responses are required, it may demand faster inference speeds, affecting computational costs

1.2.5. Resource Utilization

GPU vs. CPU: Training and inference on GPUs are generally faster than on CPUs, but GPUs may come with higher associated costs.

Parallelization: Models designed for parallel processing can take advantage of multiple GPUs or distributed computing, affecting computational efficiency.

1.2.6. Model Optimization

Quantization: Techniques like quantization (reducing precision of weights) can be used to optimize models and reduce computational costs.

Model Pruning: Pruning techniques can be applied to reduce the number of parameters, lowering computational requirements.

1.2.7. Hardware Infrastructure

Cloud vs. On-Premises: Using cloud services for model training and inference may offer flexibility but can come with associated costs.

Specialized Hardware: Some models may benefit from specialized hardware, such as TPUs (Tensor Processing Units) or ASICs (Application-Specific Integrated Circuits), which can impact costs.

As technology evolves, more efficient algorithms and hardware solutions may emerge, influencing the computational cost landscape for CQA models. Additionally, considerations for environmental sustainability may drive the development of more energy-efficient models [7,8,9].

2. Related Work

To design the proposed system we carried out survey of various approaches being used for the system.

Author Mohini Wakchaure Et. Al. [1] believes that the CQA system for public question answering requires a semantic gap between pairs of questionnaires is a major problem in the QA system and contextual modeling. The author has made sense of a deep neural network with three layers namely; Convolution Neural Network, Long-Term Memory and a Randomized Conditions. To test the SemEval-2015 CQA database is used. Attentive Neural Network architecture is used to develop a CQA system. The bi-LSTM monitor is used to upgrade which is a unit of Recurrent Neural Network

In this paper, author Oumayma Chergui Et Al. introduce a novel solution for semantic textual similarity that leverages a two-fold approach. Firstly, we adopt a semantic graph-based strategy to account for the varying semantic importance of terms within questions. Secondly, we incorporate Bayesian inference to handle the inherent semantic uncertainty present in natural language texts. Our proposed solution is integrated into an educational Community Question Answering (CQA) system, utilizing an archive of questions and answers from a community of students. The core functionality of our approach revolves around measuring similarity between a new question and previously answered questions. By employing this similarity measure, we aim to retrieve the most relevant and similar solutions from the existing knowledge base for the new problem. While our approach is specifically tailored for the context of CQA systems, we design it as a versatile solution applicable to a broader spectrum of text mining applications requiring textual similarity measures, especially for short natural-language texts. The semantic graph, a key component of our approach, has potential applications in query expansion, keywords extraction from texts, annotating short texts, and beyond. Furthermore, given that our approach focuses on question–question similarity, it opens avenues for additional applications in CQA systems and question answering more broadly. These potential applications include question recommendation, question duplicate detection, and other areas where understanding and measuring similarity between questions are crucial for system effectiveness [2].

Author Ankur Pan Saikia developed an innovative hybrid approach, leveraging TRIE and Semantic Matching algorithms that can significantly enhance expertise identification in these systems. The TRIE data structure proves to be a powerful tool for efficiently indexing user profiles and questions, enabling rapid retrieval of users with expertise in specific domains. Simultaneously, Semantic Matching algorithms employ advanced natural language processing techniques to assess the alignment between the content of a question and a user's expertise.

Table 1. Analysis of Various Models Used in CQA

Model	*Application in CQA*	*Advantages*	*Disadvantages*
Support Vector Machines	Answer ranking,	Effective for categorizing	IMay struggle with capturing intricate
(SVM)	question classification	and ranking questions	semantic relationships in text

Naive Bayes	Answer classification, sentiment analysis	Simple and computationally efficient	Assumes independence between words, may oversimplify language nuances
Random Forests / Decision Trees	Question categorization, answer extraction	Interpretable, handles non-linearity well	May struggle with capturing complex linguistic patterns
Hidden Markov Models (HMM)	Part-of-speech tagging in questions	Suitable for sequential data, interpretable	May struggle with capturing long-range dependencies in language
Conditional Random Fields (CRF)	Named entity recognition in questions	Models dependencies between adjacent labels	Computationally more expensive than simpler models
Word2Vec, GloVe	Similarity-based answer retrieval	Captures semantic relationships in word vectors	May not handle the intricacies of community-specific language well
FastText	Similarity-based answer retrieval	Handles out-of-vocabulary words, character n-grams	Limited understanding of long-range dependencies
Ensemble Models (AdaBoost, Gradient Boosting)	Answer ranking, question classification	Combines multiple weak learners for robustness	Can be computationally expensive for large datasets
Regular Expressions	Pattern matching for question extraction	Simple, efficient for specific pattern matching	May lack flexibility for capturing diverse question patterns
TF-IDF (Term Frequency-Inverse Document Frequency)	Answer extraction, relevance ranking	Represents word importance in a document corpus	Ignores word order and semantic relationships in a sequence

Bag-of-Words (BoW)	Answer classification, topic modeling	Simple representation of word occurrences	Ignores word order and semantic relationships in a sequence

Techniques such as keyword matching, semantic similarity analysis, and topic modeling are employed to scrutinize user profiles and questions. The synergistic application of TRIE-based expertise identification and Semantic Matching algorithms enables the identification of the most suitable expert for a given question. This integrated approach contributes to the improvement of answer quality and user satisfaction. In summary, the adoption of a hybrid approach involving TRIE and Semantic Matching algorithms empowers CQA systems to effectively pinpoint experts, ultimately elevating the overall quality of provided answers. This innovative strategy aligns with the goal of enhancing user experiences within the community-driven question answering environment [3].

Taihua Shao Et al. [4] proposed a collaborative learning model to select answers in response to the question, aimed at introducing a rich feature of embedded sentence. The author has adopted a coherent structure to integrate sentence embedded generated by different networks to eliminate the shortcomings of a single neural network in the representation of distributed sentences. Here the author combines the WR model with neural networks. In order to test the proposed framework, the author conducted experiments on the freely available QA database, i.e., Insurance QA. Test results show that the proposed learning methods work better than competing foundations in terms of known test metrics.

Table 1 provides a comprehensive analysis of diverse models employed in Community Question Answering (CQA) systems, evaluating their applications, advantages, and disadvantages.

3. Research Methodology

Proposed CQA model functions as a natural language processing (NLP) application with dual goals. Firstly, it aims to comprehend the intricacies involved in natural language processing and its representation. Secondly, it strives to enhance communication in natural language in minimum computational cost.

Illustrated in Fig. 2 is the dynamic configuration of the system structure. Within this chapter, we delve into the categorization of this program into three pivotal modules, namely:

User Question/Query Processing

Qtag Based Dataset Question Retrieval & Processing

Question Ranking & Answer Extraction

Answer Ranking

3.1. User Question/Query Processing

Handling input queries and dataset questions involves a structured approach. In this context, the input query undergoes a pre-processing phase aimed at simplifying the intricacies associated with extracting comparable features when evaluating semantic relationship with dataset questions. Within this module, three distinct pre-processing methods are utilized to streamline this process.

3.1.1. Tokenization

Tokenization involves breaking down the input query and dataset questions into individual tokens, which are typically words or subwords. This step simplifies the representation of the text, making it easier to compare and analyze. For example, the sentence "Processing input questions" would be tokenized into ["Processing", "input", "questions"].

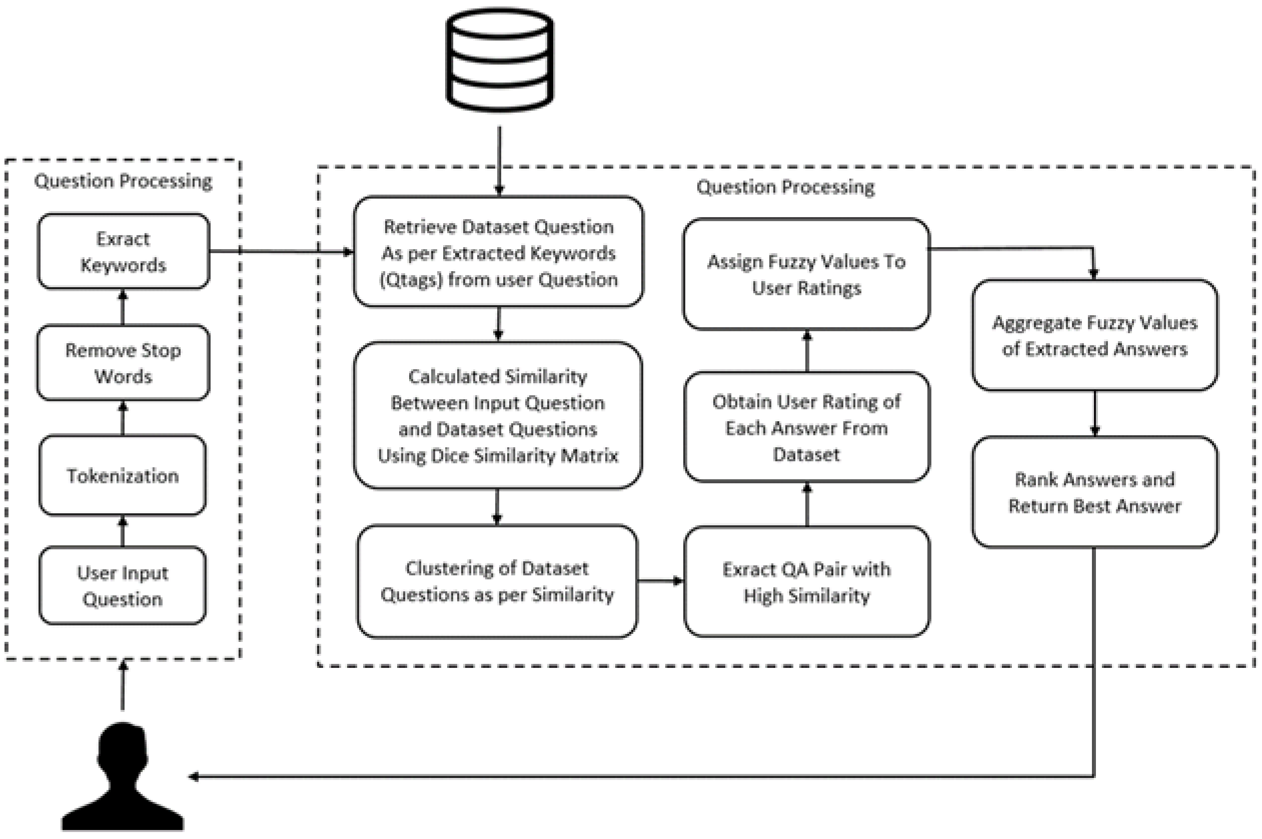
Question Processing
Exract Keywords
Remove Stop Words
Tokenization
User Input Question
Question Processing
Retrieve Dataset Question As per Extracted Keywords (Qtags) from user Question
Calculated Similarity Between Input Question and Dataset Questions Using Dice Similarity Matrix
Clustering of Dataset Questions as per Similarity
Exract QA Pair with High Similarity
Obtain User Rating of Each Answer From Dataset
Assign Fuzzy Values To User Ratings
Aggregate Fuzzy Values of Extracted Answers
Rank Answers and Return Best Answer

Fig. 2. Proposed CQA Architecture

3.1.2. Stop Word Removal

Stop words, which are common words like "the," "and," or "is," may not contribute significantly to the meaning of a sentence. Removing these stop words can help focus on the essential content of the text and reduce noise in the comparison process. This step is beneficial for improving the efficiency of similarity calculations.

```
Algorithm: Query Processing
Data: String Question
Result: String[] words
1. Tokenization:
   Split the input question into individual words.
     String[] words = Question.split("\\s+")
2. Remove Stop Words:
   For each word in the array:
     Check if the word is not empty.
       If the word is a stop word, remove it.
         For Each (word in words)
           if (word.isEmpty()) continue
           if (isStopword(word))
           Remove(word)
         End For
   End For
```

```
3. Stemming:
    For each word in the array:
     Check if the word is not empty.
     If the stemmed version of the word is a stemmed stop word, remove it.
     For Each (word in words)
        if (word.isEmpty()) continue
        if (isStemmedStopword(word))
           Remove(Stem)
    End For
4. Return Result:
Return the modified array of words.
   Return words
```

3.1.3. Stemming

Stemming and lemmatization are methods to reduce words to their base or root form. Stemming involves removing prefixes or suffixes to obtain the word's stem (e.g., "running" becomes "run").

3.2. Qtag Based Dataset Questions Retrieval & Processing

In our approach, we leverage a similarity-based method to find the connection between user input questions and dataset queries, yielding a set of candidate questions. This process incorporates the Qtag approach, employing an analogy-based technique to identify relationships between user queries and those within the dataset, thereby generating a relevant set of queries.

3.3. Calculating String Similarity

The string similarity module aims to assess the edit distance between an input question and collection of dataset questions, retrieved through Qtag Based Dataset Questions Retrieval. The methodology employed leverages a string similarity matrix to precisely measure the distance between the user's question and the dataset questions..

For computing the similarity score between two sentences (questions), the application of the following formula is integral:

$$word\_similarity\_score = \frac{avg.string\ length - distance}{avg.string\ length} \quad (1)$$

Example

What are the symptoms of diabetes? (length = 6)

How is diabetes diagnosed and treated? (length = 6)

Table 2. Similarity Index Matrix

	What	*Are*	*The*	*Symptoms*	*Of*	*Diabetes*
How	0	0	0	0	0	0
Is	0	0	0	0	0	0
diabetes	0	0	0	0	0	1
diagnosed	0	0	0	0	0	0
and	0	0	0	0	0	0
treated	0	0	0	0	0	0

$$similarity\_score = similarity\_index/avg.string\ length \quad (2)$$

0.16 = 1/6/2

3.4. Question Ranking & Answer extraction module

In this adaptation of the K-Means clustering algorithm, a departure from the conventional use for classification is made. Typically, clustering algorithms are employed solely for classification and creation of clusters based on specific criteria. However, in our modification, the primary objective is to facilitate the ranking of sentences within a cluster. This is a unique approach to the K-Means clustering algorithm.

In this module, questions having lower edit distances are organized into clusters. The process involves generating a hash table to store sentences sharing similar similarity scores. Subsequently, this hash table is arranged in descending order. The mean value, denoted as 'K,' is computed from the similarity scores within the hash table. Sentences with scores surpassing the mean value are allocated to the first set, whereas those with scores below the mean value are assigned to the second set. This systematic approach results in the creation of a collection showcasing a high level of similarity, with sentences reorganized in descending sequence. The modification is designed to adapt the K-Means algorithm specifically for the task of ranking sentences within clusters based on similarity metrics.

```
Algorithm: Calculating String Similarity
Data: User_Question, String Dataset_Question
Result: Distance d
1. Tokenization:
    Split the user question and dataset question into individual words.
        String[] user_words = User_Question.split("\\s+")
        String[] dataset_words = Dataset_Question.split("\\s+")
2. Remove Stop Words:
    Remove stop words from each array
        RemoveStopWord(user_words).
        RemoveStopWord(dataset_words).
3. Calculate String Distance::
    Calculate the string distance between the processed arrays using a String Similarity function.
  Check if the word is not empty.
    If the stemmed version of the word is a stemmed stop word, remove it.
        d = StringSimilarity(user_words, dataset_words)
Return Result:
Return the calculated distance.
    Return d
```

Above algorithm outlines the process of calculating string similarity between a user's question and a dataset question, involving tokenization, stop word removal, and the computation of string distance. The result is the distance value indicating the similarity between the two questions.

Algorithm: Clustering

Data: HashMap<String,Double> *hm*

Result: Cluster c1, c2

```
1. For Each (i to hm.size()):
       Add question similarity scores.
           score += hm.getValue()
   End For
2. Calculate Mean Value for All Questions:
       Remove stop words from each array
           Mean Value = score / hm.size()
           RemoveStopWord(dataset_words).
3. Sort HashMap in Descending Order:
       Sort the HashMap in descending order based on
       similarity scores.
   For Each (entry in sorted HashMap):
       If (entry.getValue() >= Mean Value)
           C1.Add(entry.getKey())
       Else
           C2.Add(entry.getKey())
   End For
```

Above algorithm performs clustering based on the similarity scores calculated for each question. It calculates the mean value of the scores, sorts the HashMap in descending order, and then assigns questions to Cluster C1 or C2 based on whether their similarity score is greater than or equal to the mean value.

3.5. Answer Extraction

In the modified clustering algorithm, the first cluster now contains questions with the least edit distance which is arranged in descending order. To obtain the answer for the highest-ranked question within this cluster, the question number is employed. Question number serves as a unique identifier assigned to each question along with its respective answer in the dataset. By utilizing the question number associated with the top-ranked question in the first cluster, the algorithm can efficiently retrieve the corresponding answer from the dataset. This integration of question number streamlines the process of linking questions to their respective answers, providing a direct and effective means of retrieving the most relevant information for the user's query within the highest-ranked cluster.

Algorithm: Answer Extraction

Data: HashMap<String, related_id> hm

Result: List<String> Answer

```
1. Get Related IDs:
      For each entry in the HashMap, retrieve the related_id.
    For Each (entry in hm):
          Related_id = entry.getValue()
  End For
2. Search for Answer with Similar Related IDs:
  For Each (related_id in hm.values()):
      String ans = SearchAnswer(related_id)
      Answer.Add(ans)
End For
3. Return Result:
      Return the list of extracted answers.
  Return Answer
```

This algorithm outlines the process of extracting answers based on related_ids stored in a HashMap. It iterates through the HashMap, retrieves the related_id for each entry, searches for the corresponding answer using the related_id, and compiles the answers into a list. The final result is the list of extracted answers.

3.6. Answer Ranking

Within this module, we have used the concept of Triangular Fuzzy Numbers (TFN) coupled with expert ratings to articulate evaluations provided by experts. In this context, a triplet (v1, v2, v3) is employed to represent the Triangular Fuzzy Number, where v1 signifies the smallest conceivable value, v2 denotes the most probable value, and v3 signifies the highest attainable value. This approach enables a nuanced representation of expert ratings and enhances the expressiveness of the evaluation process.

Figure 3 represents the corresponding membership function for the variables shown in Table 3, and this shows the degree of membership of each class of linguistic term.

Table 3. Linguistic variables and associated TFN

Sr. No.	*Linguistic Term*	*TFN*
1	Low Weight	(0,1,2)
2	Medium Weight	(1,2,3)
3	Normal Weight	(2,3,4)
4	High Weight	(3,4,5)

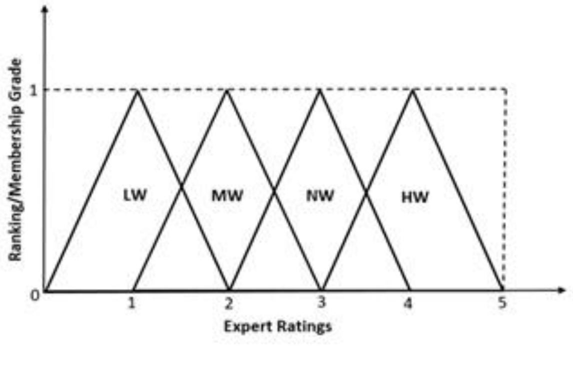

Fig. 3. Membership functions of triangular fuzzy numbers

Here the question, answer and expert maintains a hierarchical relationships among them. One Question can have multiple answers and one answer can have multiple expert's ratings. So for every answer to a particular question an aggregate value is obtained from all the experts

4. Result Analysis

In our result evaluation, we employed the HealthTap dataset, an extensive collection comprising over 40,000 questions and corresponding answers from the online health portal. HealthTap serves as a valuable resource for obtaining medical advice.

Fig. 4 illustrates a comparative analysis of two systems: one implementing a clustering algorithm and another operating without clustering, with a focus on time efficiency. The time measurements, depicted in milliseconds, showcase that the system incorporating the clustering algorithm outperforms the non-clustering system.

The X-axis denotes the names of the systems, while the Y-axis represents time measured in seconds. This analysis emphasizes the efficiency gains achieved by incorporating the clustering algorithm, demonstrating its superiority in terms of processing time when compared to the system without clustering.

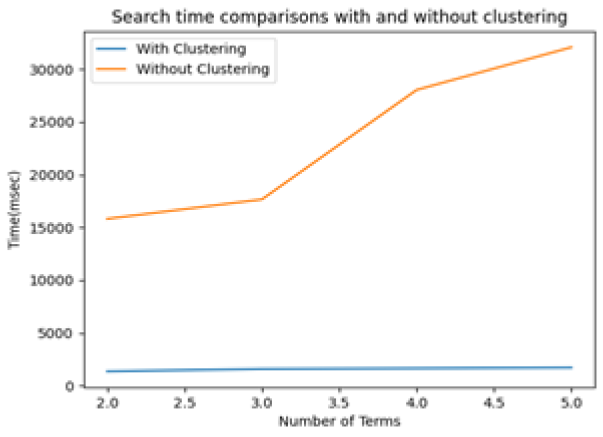

Fig. 4. Time Comparison with and without clustering

Fig. 5 shows the term wise system performance graph of the proposed system. Above In figure X-axis shows the different term count and Y-axis shows the time required for searching related question.

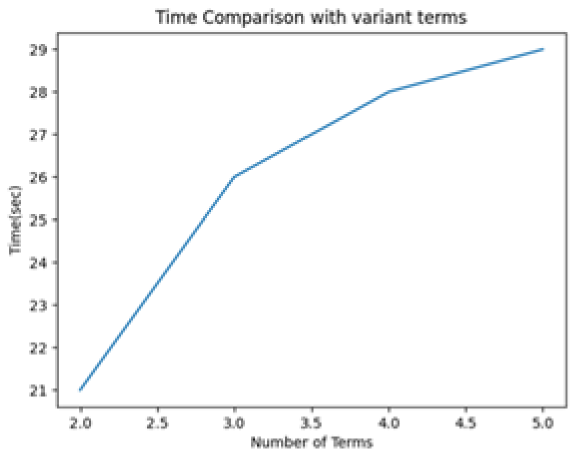

Fig. 5. Time Comparison with variant terms

Question: I am suffering from knee pain from 2 months. is it symptoms of bone cancer.

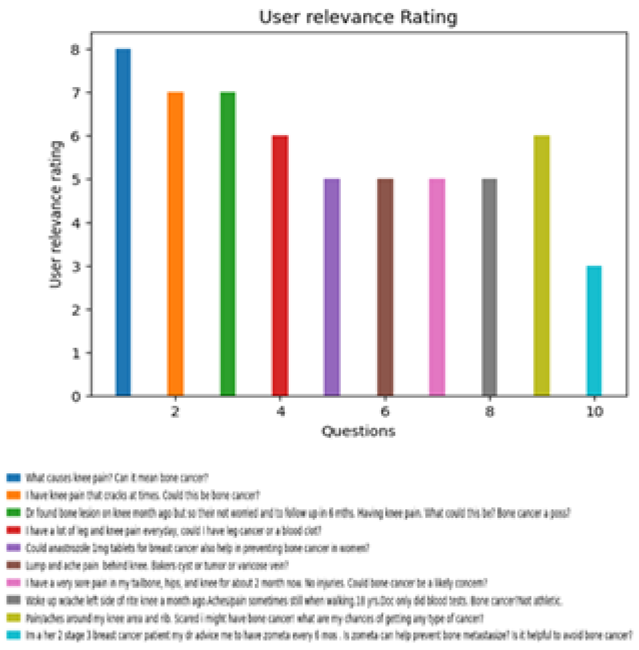

Fig. 6. Question set 1 with relevance votes

Question: I am having chest pain from 2 days what does that mean?

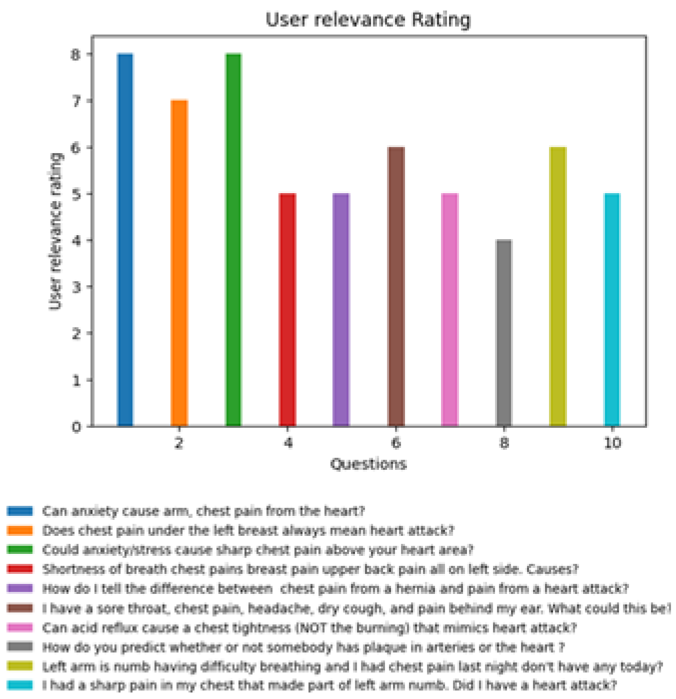

Fig. 7. Question set 2 with relevance votes

Fig. 6, 7 shows the question wise relevance voting graph. Above In figure X-axis shows the different questions relevant to user input question and Y-axis shows the relevance votes for each question

5. Conclusion

The research endeavours to not only enhance the effectiveness of public questionnaires through content-driven responses but also emphasizes the importance of cost efficiency in the proposed model. The study investigates the fundamental aspects of term selection and sentence sorting, comparing their efficiency at various levels of depth. An exhaustive analysis of words, phrases, and sentence relationships is conducted to assess model performance, with a particular focus on cost-effective methodologies.

Experiments are designed based on term-based methods integrated with Triangular Fuzzy Numbers, with an additional consideration for cost-efficient implementation. Parameters such as system accuracy and cost-effectiveness are utilized to comprehensively evaluate the performance of the system. The proposed question-answering framework strategically employs cosine similarity and a ranking algorithm to achieve timely outputs, ensuring not only prompt but also economically viable responses to public inquiries. The integration of cost efficiency underscores the practical applicability and sustainability of the proposed system in real-world scenarios.

Acknowledgements

I would like to express my sincere gratitude to my guide, Mr. Neeraj Sahu, for his invaluable support and guidance throughout the research and development of the paper. His expertise, insightful feedback, and unwavering encouragement have been instrumental in shaping the direction and quality of this work. I am truly thankful for his mentorship, which has been a source of inspiration and motivation.

Author contributions

Manisha Khadse and **Dr. Neeraj Sahu** made significant contributions to this work. **Manisha Khadse** highlights the key features of the proposed model that contribute to its success, particularly the streamlined identification of similar questions and the scoring mechanism based on both question similarity and expert ratings. By emphasizing the precision and timeliness of answers in the CQA system, which reinforces the practical significance of the introduced recommendation system. The inclusion of the HealthTap dataset in the evaluation adds credibility to the research findings, showcasing the model's ability to address real-world scenarios and offering insights into its potential deployment in diverse contexts. Overall, contribution strengthens the research by providing empirical evidence of the proposed model's effectiveness and its practical superiority over existing CQA systems.

Dr. Neeraj Sahu significantly contributes to the research by identifying and addressing a critical issue in Community Question Answering (CQA) systems like the time-consuming process of sorting relevant questions. The primary focus of their contribution lies in devising a recommendation system aimed at enhancing the efficiency of CQA systems. By introducing the innovative concept of Question Tags (Qtags), Qtags not only aid in finding analogous questions but also contribute to determining the domain of a given question, thereby adding a layer of contextual understanding.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Mohini Wakchaure, Prakash Kulkerni, India "A Scheme of Answer Selection In Community Question Answering Using Machine Learning Techniques" International Conference on Intelligent Computing and Control Systems IEEE (ICICCS 2019),879-883

[2] Chergui, O., Begdouri, A., & Groux-Leclet, D. (2019). "Integrating a Bayesian semantic similarity approach into CBR for knowledge reuse in Community Question Answering. Knowledge-Based Systems", *104919.* doi:10.1016/j.knosys.2019.104919

[3] Ankur Pan Saikia, "Enhancing Expertise Identifcation in Community Question Answering Systems (CQA) Using a Hybrid Approach of TRIE and Semantic Matching Algorithms", Annals of Multidisciplinary Research, Innovation and Technology (AMRIT), 2(1), 2023, 27-34

[4] TAIHUA SHAO , XIAOYAN KUI , PENGFEI ZHANG, AND HONGHUI CHEN 1Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China "Collaborative Learning for Answer Selection in Question Answering" 2169-3536 , 2018 IEEE.

[5] Darshana V. Vekariya, Nivid R. Limbasiya, Computer Engineering Department V.V.P. Engineering College, Rajkot Gujarat, India "A Novel Approach for Semantic Similarity Measurement for High Quality Answer Selection in Question Answering using Deep Learning Methods", 978-1-7281-5197-7/20/$31.00 ©2020 IEEE

[6] WEIJING WU1,YANG DENG, YUZHI LIANG, AND KAI LEI.ICNLAB, School of Electronics and Computer Engineering (SECE), Peking University, Shenzhen, China, The Chinese University of Hong Kong, Hong Kong, China "Answer Category-Aware Answer Selection for Question Answering", DOI10.1109/ACCESS.2020.3034920, IEEE

[7] Issa Annamoradnejad , Mohammadamin Fazli , Jafar Habibi , Department of Computer Engineering Sharif University of Technology Tehran, Iran , "Predicting Subjective Features from Questions on QA Websites using BERT", 978-1-7281-1051-6/20/$31.00 ©2020 IEEE

[8] Xianming Yao, Jian Xu, Wanhua Yang,"Study on Chinese Open Domain Question Answering based on Support Vector Machine", School of Information Engineering QuJing Normal University Qujing, China, 978-1-7281-7008-4/20/$31.00 ©2020 IEEE

[9] Yongliang Wua, Shuliang Zhao, School of Mathematical Sciences, Hebei Normal University,

Hebei 050024, China “Community answer generation based on knowledge graph”, 2020 Elsevier

[10] TAIHUA SHAO , YUPU GUO, HONGHUI CHEN, AND ZEPENG HAO, Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China, “Transformer-Based Neural Network for Answer Selection in Question Answering”, 2169-3536 , 2019 IEEE

[11] Jamshid Mozafari, MohammadAli NematBakhsh, Afsaneh Fatemi Department of Software Engineering, University of Isfahan, Isfahan, Iran “Improved Answer Selection For Factoid Questions”, 978-1-7281-5075-8/19/$31.00 ©2019 IEEE

[12] Juee Gosavi , B. N. Jagdale , MIT College of Engineering, Pune, India “Answer Selection In Community Question Answering Portals”, 2018 IEEE

[13] Tirath Prasad Sahu, Reswanth Sai Thummalapudi, Naresh Kumar Nagwani, “Automatic Question Tagging Using Multi-label Classification in Community Question Answering Sites”, *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom),* 03 October 2019 IEEE, DOI: 10.1109/CSCloud/EdgeCom.2019.00-17

[14] Jian Song, Xiaolong Xu, Xinheng Wang, “TSAR-based Expert Recommendation Mechanism for Community Question Answering”, *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD),* 2021 IEEE, DOI: 10.1109/CSCWD49262.2021.943