Exploring Jaccard Similarity and Cosine Similarity for Developing an Assamese Question-Answering System

Chapter · November 2023

DOI: 10.1007/978-981-99-5881-8_8

Nomi Baruah
Dibrugarh University
35 PUBLICATIONS **95** CITATIONS
SEE PROFILE

Rituraj Phukan
Dibrugarh University
10 PUBLICATIONS **1** CITATION
SEE PROFILE

Exploring Jaccard Similarity and Cosine Similarity for Developing an Assamese Question Answering System

Nomi Baruah[1], Saurav Gupta[1] Subhankar Ghosh[1], Syed Nazim Afrid[1], Chinmoy Kakoty[1], and Rituraj Phukan[1]

Dibrugarh University, Dibrugarh 786004, India
baruahnomi@gmail.com, souravmusic84@gmail.com,
ghosh.subhankar619@gmail.com, nazimafrid99@gmail.com,
chinmoykakoty20@gmail.com, riturajphukan01@gmail.com

Abstract. The Assamese Question Answering System(AQAS) is an important Machine Learning (ML) approach that aids a user in finding relevant information using Assamese Natural Language Processing. We used three mathematical and statistical approaches for AQAS based on data from question responses in this research article. Cosine similarity and the Jaccard similarity algorithm are two of these approaches. On user questions and questions responding to data, the cosine similarity has interacted with Non-negative Matrix Factorization (NMF) to lower the space and time complexity. Preprocessing data and establishing a relationship between user queries and contained instructive questions are the two elements of this study's methods. It is found that we have got an accuracy score of 93.8% for cosine similarity and 87.38% for Jaccard similarity. It is found that Cosine similarity outperforms Jaccard similarity by 6.42%.

Keywords: AQAS, Information retrieval, Machine Learning, Mathematics, Statistics

1 INTRODUCTION

The information age is currently underway. As the amount of information available grows and the world becomes more informational, the virtual information retrieval system, which is an artificial question-answering system, maintains its importance [1]. Users frequently have specific questions in mind, which is why they seek responses. They want answers that are simple and explicit, and they always want to ask inquiries in their local language rather than being limited to a query language, query formation rules, or even a certain knowledge domain. The new method to match user demands is to do a linguistic study of the question and seek to grasp what the user really means [2].

The AQAS is made up of three primary modules in Assamese NLP: data collecting, information and user question processing, and establishing relationships between them [3]. For the purpose of NLP preprocessing, approaches used are Tokenization and Stop Word Removal. We have used the NLTK Tokenizer that divides strings into lists of substrings. We employed the NMF to lower the dimension of a question and information while simultaneously reducing the program's execution time [4]. It also makes it easier to comprehend and calculate in a straightforward manner.

We employed Cosine similarity and Jaccard similarity to produce responses to user inquiries. These techniques aid in the establishment of connections between users' questions and information.

The contributions are summarized as follows:

- We have introduced mathematical and statistical procedures for AQAS for information retrieval.
- For the pre-processing of data, we have done Stop Word Removal and Tokenization.
- In order to generate answers we used Cosine similarity and Jaccard similarity.
- We have used the NMF with cosine similarity to reduce time and space complexity as well as the instant answering of questions.

The rest of the paper is organized as follows: Sect. 2 provides the related work and study. Section 3 explains the background study for the project, which was followed by Section 4, which consists of our proposed work. Section 5 presents the pre-processing stage of the project. Section 6 contains the establishment and relationship between the two algorithms that are being used. Section 7 provides the experiment and results of the project and lastly, Section 8 explains the conclusion and future work of the project.

For the easiest explanation, we consider two users questions as examples in every term of this paper, these are:

User Question-1: অসমত আটাইতকৈ ডাঙৰ মহিলা হল ক'ত অৱস্থিত? [Where is the largest female hall situated in Assam?]

User Question-2 : ইয়াৰ প্ৰেক্ষাগৃহৰ নাম কি? [What's the name of its auditorium?]

2 RELATED STUDY

Question-Answering(Q-A) systems, as stated in the introduction, are a good option for textual information retrieval, knowledge sharing, and discovery. This is why a huge number of Q-A systems have lately been developed [5]. These systems cater to a wide range of international languages. However, some are better served than others. Furthermore, large-scale Q-A techniques for a much larger environment, such as the World Wide Web, have received significant attention. Some search engines include a question-answering feature that works well [6].

Many different Q-A systems have been developed in various settings when it comes to languages. Q-A systems have been extensively examined in the instance of Latin languages. Particularly well-served is English. This is primarily due to the fact that the vast majority of documents on the internet are written in English. "Base-ball" is one of the oldest question-answering systems. It supports a finite number of questions on corpora having a defined collection of documents [7]. The web is used as a resource by Q-A systems, such as "start" and "swingly", which both employ search engines to get answers. There is also "qalc", a Q-A system for

English factoids in the free domain. For each query, our system does a syntactic and semantic analysis. Nonetheless, it has some faults as a result of insufficient syntactical rules. (Edipe is a morpho-syntactic pattern-based Q-A system created by the LIC2M in France.) However, in terms of the tools employed, it takes a minimalist approach [8].

Gomes et al. propose a Hereditary Attentive Template-based Approach for Complex Knowledge Base Question Answering Systems (KA-HAT), which combines the advantages of templates and deep learning to enhance the accuracy of question answering [9]. The proposed approach achieves an F1 score of 82.5% on a benchmark dataset. Do et al. presents a BERT-based triple classification model that uses knowledge graph embeddings for question answering [10]. The model achieves an accuracy of 84.2% on the WebQuestionsSP dataset, outperforming several state-of-the-art models. Sentiment analysis [11] is being used for improving the performance of question-answering systems. There are a lot of research works that focus on utilizing sentiment analysis to improve the performance of question-answering systems [12, 13]. These approaches show promising results in predicting the sentiment and topic of queries and selecting the best answer in CQAS, respectively. The accuracy achieved by the systems demonstrates the potential of sentiment analysis in improving the performance of Q-A systems. Short-text clustering [14] is also used in developing Q-A systems [15].

The future looks bleak for the Assamese language. It is, nonetheless, one of the top ten languages on the internet. In general, the Assamese language lacks computerized tools and resources. Q-A systems created exclusively for Assamese are one type of these missing tools [16]. The number of developed Assamese Q-A systems is still small as compared to those established for English or French, for example. This is owing to two factors: a lack of access to linguistic resources and tools, such as corpora and fundamental Assamese NLP tools, and the language's extremely complicated character (for instance, Assamese is inflectional and not concatenative and there is no capitalization as in the case of English) [5].

3 BACKGROUND

3.1 Tokenization

Tokenization is the process of breaking down large pieces of text into smaller ones. Tokenization divides the raw text into words and sentences, which are referred to as tokens. These tokens aid in the comprehension of the context or the development of the NLP model. By evaluating the sequence of words, tokenization aids in interpreting the meaning of the text. We have used the NLTK Tokenizer that divides strings into lists of substrings.

3.2 Stop-word Removal

In computing, stop words are words that are filtered out before or after the natural language data (text) are processed. Stop word removal is one of the most commonly used preprocessing steps across different NLP applications. The idea is simply to remove the words that occur commonly across all the documents in the corpus. Typically, articles and pronouns are generally classified as stop words.

3.3 Cosine Similarity

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of two non-zero vectors as mentioned in [17] can be derived as:

$$A \cdot B = ||A|| \cdot ||B|| \cdot cos\theta \tag{1}$$

3.4 Jaccard Similarity

The Jaccard index also called the Jaccard similarity, is a statistical method for determining the similarity of distinct sample sets. If P and Q be two sets the Jaccard index formula as mentioned in [17] is given:

$$J_{index}(P,Q) = \frac{|P \cap Q|}{|P \cup Q|} * 100 = \frac{|P \cap Q|}{|P| + |Q| - |P \cup Q|} * 100 \tag{2}$$

4 PROPOSED WORK

In this research paper, we have presented an Assamese Question Answering System using Assamese Natural Language Processing. The procedure is isolated into three parts that are: informative document collection, pre-processing data, and relationships between information and user questions. The action of Cosine Similarity and Jaccard similarity algorithms are urged to obtain the relationship between the questions and answers.

A group of question-and-answer pairs collected on a specific topic is referred to as an informative document collection. Following the user's question, Cosine or Jaccard algorithms will be used to determine the relationship between the user's question and the questions in the dataset. The answers to the questions with the most similarity will be returned.

The user will provide input in the form of a question. The question will then be pre-processed, which means it will be cleansed to see if there is any undesirable data. Following that, tokenization and stop-word removal will be performed. The training dataset's questions will also be pre-processed in the same way. Following preprocessing, the Jaccard and Cosine similarity algorithms will be used to compare the query question and the dataset questions for similarity. All of the questions in the training dataset will be checked for similarity by the algorithms. The user will receive the answer connected with the question that has the greatest similarity.

Figure 1 shows a pictorial diagram of the proposed work.

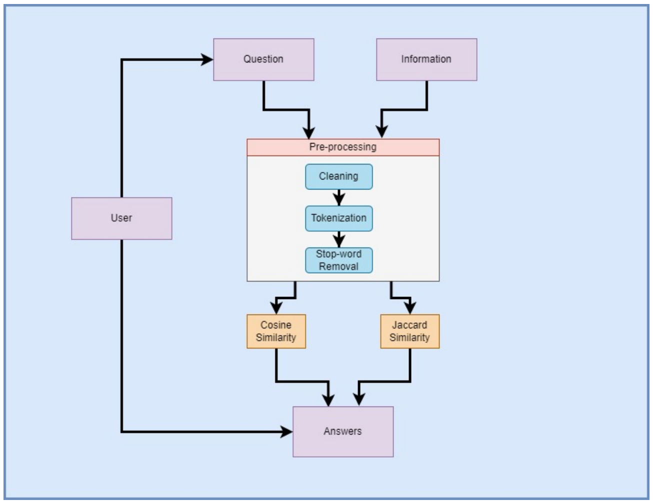

Fig. 1. Pictorial diagram of the proposed work

5 PRE-PROCESSING

The dataset must be preprocessed to run the algorithm. There are two preprocessing techniques held in the Assamese Question Answering System. Cleaning words refers to removing an unwanted character that does not have any sentiment on informative data; for example colon, semicolon, comma, question mark, exclamation point, and other punctuations.

Word Tokenization is the most commonly used tokenization algorithm. It splits a piece of text into individual words based on a certain delimiter. Depending upon delimiters, different word-level tokens are formed.

For example: প্ৰথমে ব্যক্তিজন তেওঁৰ ঘৰলৈ গৈছিল | (prathame byaktijon teor ghorloi goisil).

After applying tokenization:

[প্ৰথমে, ব্যক্তিজন, তেওঁৰ, ঘৰলৈ, গৈছিল, |]

Now stop word removal. Stop words refer to the words that do not have any influence on documents or sentences, but help to complete the sentence. The second one is stop word removal. After removing the stop words the tokens left are: [ব্যক্তিজন, তেওঁৰ, ঘৰলৈ, গৈছিল]

6 ESTABLISHMENT OF RELATIONSHIP

6.1 Jaccard Similarity

The mathematics behind the Jaccard Similarity algorithm has been displayed in the following example. Here initially we took two sentences as doc_1 and doc_2. Later we took the intersection and union of the two sentences as expressed in equation no. 3.

After pre-processing, the questions and data can be revealed as,
doc_1= ”মই বাইকেৰে কলেজলৈ গৈ আছো”

doc_2 = ”মই কলেজলৈ গৈ আছো”

Let’s get the set of unique words for each document.
$words\_doc1$ = ’মই’, ’বাইকেৰে’, ’কলেজলৈ’, ’গৈ, ’আছো’

$words\_doc2$ = ’মই’, ’কলেজলৈ’, ’গৈ’, ’আছো’

Now, we will calculate the intersection and union of these two sets of words and measure the Jaccard Similarity between doc_1 and doc_1.

$$J(doc\_1, doc\_1) = \frac{\text{(’মই’, ’বাইকেৰে’, ’কলেজলৈ’, ’গৈ, ’আছো’)} \cap \text{(’মই’, ’কলেজলৈ’, ’গৈ’, ’আছো’)}}{\text{(’মই’, ’বাইকেৰে’, ’কলেজলৈ’, ’গৈ, ’আছো’)} \cup \text{(’মই’, ’কলেজলৈ’, ’গৈ’, ’আছো’)}} \tag{3}$$

$$= \frac{\text{(’বাইকেৰে’)}}{\text{(’মই’, ’বাইকেৰে’, ’কলেজলৈ’, ’গৈ, ’আছো’)}}$$

$$= \frac{1}{5} = 0.2$$

6.2 Cosine Similarity

Cosine similarity is one of the metrics to measure the text-similarity between two documents irrespective of their size in Natural language Processing. A word is represented in a vector form. The text documents are represented in n-dimensional vector space.

Mathematically, the Cosine similarity metric measures the cosine of the angle between two n-dimensional vectors projected in a multi-dimensional space. The Cosine similarity of the two documents will range from 0 to 1. If the Cosine similarity score is 1, it means two vectors have the same orientation. A value closer to 0 indicates that the two documents have less similarity.

The mathematical equation of Cosine similarity between two non-zero vectors is:

$$similarity = cos(\theta) = \frac{A.B}{||A||.||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4}$$

Let’s see the example of how to calculate the cosine similarity between two text documents:
doc_1 = ”ডাটা হৈছে ডিজিটেল অৰ্থনীতিৰ তেল”

doc_2 = ”ডাটা হৈছে এটা নতুন তেল”

Vector representation of the document

doc_1_vector = [1, 1, 1, 0, 1, 0, 1]

doc_2_vector = [1, 0, 0, 1, 1, 1, 1]

$A.B = \sum_{i=1}^{n} A_i B_i = (1{*}1)+(1{*}0)+(1{*}0)+(0{*}1)+(1{*}1)+(0{*}1)+(1{*}1) = 3$

$\sqrt{\sum_{i=1}^{n} A_i^2} = \sqrt{1+1+1+0+1+0+1} = \sqrt{5}$

$\sqrt{\sum_{i=1}^{n} B_i^2} = \sqrt{1+0+0+1+1+1+1} = \sqrt{5}$

cosine similarity = $cos\theta = \frac{A.B}{|A|.|B|} = \frac{3}{\sqrt{3}\sqrt{5}} = \frac{3}{5} = 0.6$

7 EXPERIMENTS

We have described a range of experiments to measure our proposed model and the mathematical and statistical procedures for AQAS. In this section, first, we present the questions that we target to reply to the experiments and describe the experimental setup. Then, we discuss the performance and result of our propounded work.

7.1 Data Preparation

Since in the future, this project can be used as a bot in any commercial website, we have built the dataset in context to a particular topic or domain(eg: travel agency booking, course subscription). We have prepared a dataset of 200 question-answer pairs as a training set. Also, we have prepared a dataset of 50 question-answer pairs as a test dataset.

7.2 Experimental Setup

We implemented our propounded model in Anaconda distribution with Python 3.7 programming language and executed them on a Windows 10 PC with an Intel Core i7, CPU (3.20GHz), and 8GB memory. Python is a high-level object-oriented language (OOP) that is suitable for scientific examination and tool development. We have used the Anaconda as the apportionment of Python. Anaconda creates the best stage for open-source data science which is powered by Python.

7.3 Result and Analysis

The analysis of the performance of these three methods was done by retrospective analysis by measuring the top-1 accuracy. Top-1 accuracy is the conventional accuracy in which the answer from the model must be the expected answer.

$$Accuracy = \frac{\textit{total no. of correctly classified news articles}}{\textit{total no. of news articles}} * 100 \tag{5}$$

In this study, we evaluated the performance of two similarity metrics, Jaccard similarity, and cosine similarity, for answering 250 questions. Our results showed that Jaccard similarity achieved an accuracy of 87.38%, while cosine similarity achieved an accuracy of 93.8%. Our findings indicate that cosine similarity outperforms Jaccard similarity in answering the given questions. The higher accuracy of cosine similarity can be attributed to its ability to capture the semantic similarity between words, as opposed to Jaccard similarity which only considers the overlap of words between two sentences. Our results suggest that cosine similarity is a more effective similarity metric for answering questions in our dataset. Future work could explore the use of other similarity metrics and their performance on larger datasets.

We compared our ideas to work that has been done in Bengali or Bangla as a point of comparison. We had to investigate the suggested approach in those scripts because Assamese and Bangla scripts have the fewest shared alphabets. However, there are a few tiny typographical changes. The letter r$\hat{o}$ is written as ”র” in Bengali/Bangla and as ”ৰ” in Assamese, whereas the character ”ৱ” pronounced as w$\hat{o}$ is written as ”ব” (b$\hat{o}$) in Bengali script. In addition, the character ”ক্ষ” (khyo) is missing from the Bengali script in Assamese. Interestingly, the Bengali/Bangla language, which has the Subject-Object-Verb (SOV) word order, has a script that is very similar, yet dialect differences occur in both languages. Table 1 represents the comparison of the proposed work with previously existing work.

The similarity of each sentence-level article was computed against itself and all the other articles. We know that self-similarity is always 1. So, all the diagonal values of the similarity matrix were replaced by 0 because we wanted to find the most similar document for any document besides itself. The most similar news article for a news article was computed by finding out the maximum similarity value in the similarity matrix.

Table 1. Comparison of accuracy with previous work

Algorithm	Our Accuracy	[17]
Cosine Similarity	93.80%	93.22%
Jaccard Similarity	87.38%	84.64%

The following tables show the illustration of the procedure explained above. D1, D2, D3, D4, and D5 represent sentence-level documents. Table 2 and Table 3 represent the cosine similarity values between the documents whereas Table 4 represents the highlighted values of the similar documents.

7.4 Comparison between English chatbot and AQAS

1. How to go to your office? (আপোনাৰ কাৰ্যালয়লৈ কেনেদৰে যাব লাগে ?)
   Mitsuku: Take the metro and stop at the terminus, we’re 5 minutes away from there.

Table 2. Similarity Matrix of five documents with each other

	D1	D2	D3	D4	D5
D1	1	0.2236	0.1884	0.2674	0.1722
D2	0.2236	1	0.1350	0.2450	0.2308
D3	0.1884	0.1350	1	0.1817	0.2840
D4	0.2674	0.2450	0.1817	1	0.2495
D5	0.1722	0.2308	0.2840	0.2495	1

Table 3. Replacing all diagonal values (self-similarity values) to 0

	D1	D2	D3	D4	D5
D1	0	0.2236	0.1884	0.2674	0.1722
D2	0.2236	0	0.1350	0.2450	0.2308
D3	0.1884	0.1350	0	0.1817	0.2840
D4	0.2674	0.2450	0.1817	0	0.2495
D5	0.1722	0.2308	0.2840	0.2495	0

Table 4. Finding out the most similar document on the basis of similarity value

	D1	D2	D3	D4	D5
D1	0	0.2236	0.1884	0.2674	0.1722
D2	0.2236	0	0.1350	0.2450	0.2308
D3	0.1884	0.1350	0	0.1817	0.2840
D4	0.2674	0.2450	0.1817	0	0.2495
D5	0.1722	0.2308	0.2840	0.2495	0

AQAS: মেট্ৰ লওক আৰু টাৰ্মিনাসত ৰওঁক, আমি তাৰ পৰা 5 মিনিট দূৰত আছোঁ |

2. How can I call you? (মই তোমাক কেনেকৈ কল কৰিব পাৰোঁ ?)
   Mitsuku: You can’t call me, you can only talk to me here.
   AQAS: আপুনি মোক ফোন কৰিব নোৱাৰে, আপুনি ইয়াত কেৱল মোৰ সৈতে কথা পাতিব পাৰে |
3. When can I check in? (মই কেতিয়া চেক ইন কৰিব পাৰো ?)
   Mitsuku: Anything you want.
   AQAS: আপোনাৰ সংৰক্ষণত নিৰ্ধাৰিত তাৰিখ |
4. What can you do? আপুনি কি কৰিব পাৰে ?)
   Mitsuku: Anything you want.
   AQAS- আপুনি বিচৰা যিকোনো বস্তু |

8 CONCLUSION AND FUTURE WORK

The Assamese Question Answering System is designed to provide exact responses to inquiries posed by users in natural language. Because different sources are the target of these systems, it is important to note that one of the most important features of Q-A Systems is their ability to provide exact answers. Then, without knowing the structure of the sources to be queried, the user asks a question in natural language. In the proposed work, the NMF is used to reduce the time complexity and the space complexity as it allows us to reduce the dimension of the data. The highest result achieved from our work is 93.80% for Cosine similarity.

We have described a system that handles a question in natural language and tries to provide its answer similar to a human answer. With certain improvements; the proposed system can be used for the following applications:

1. Can be used in situations where a quick review of an entire text takes time.
2. Adding speech recognition abilities to the current system will enable people with reading disabilities to take advantage of this system.
3. This system can be used to make online lectures more old-school type by allowing lectures to proceed only when questions related to the previous lecture are answered correctly.
4. The QA paradigm extends beyond AI systems to query processing in database systems and many analytical tasks that involve gathering, correlating, and analyzing information; can naturally be formulated as QA problems.
5. The existing system can be integrated with a search engine to enhance its performance.
6. The existing system can be integrated with any website as a chatbot to solve users’ queries.

In future studies, the dataset size could be increased for better accuracy. Future research of this work can explore new ways to improve the performance of the question-answering system by implementing other approaches such as neural networks, deep learning, and machine learning algorithms. Another possible direction is incorporating user feedback into the system, which can help the model learn from its mistakes and refine its predictions.

References

1. Lai, Y., Jia, Y., Lin, Y., Feng, Y., Zhao, D.: A chinese question answering system for single-relation factoid questions. In: Natural Language Processing and Chinese Computing: 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8–12, 2017, Proceedings 6, pp. 124–135. Springer (2018)
2. Sahu, S., Vasnik, N., Roy, D.: Prashnottar: a hindi question answering system. International Journal of Computer Science & Information Technology **4**(2), 149 (2012)
3. Hammo, B., Abu-Salem, H., Lytinen, S.L., Evens, M.: Qarab: A: Question answering system to support the arabic language. In: Proceedings of the ACL-02 workshop on Computational approaches to semitic languages (2002)
4. Gupta, P., Gupta, V.: A survey of text question answering techniques. International Journal of Computer Applications **53**(4) (2012)
5. Gupta, V., Lehal, G.S.: Named entity recognition for punjabi language text summarization. International journal of computer applications **33**(3), 28–32 (2011)
6. Uddin, M.M., Patwary, N.S., Hasan, M.M., Rahman, T., Tanveer, M.: End-to-end neural network for paraphrased question answering architecture with single supporting line in bangla language. International Journal of Future Computer and Communication **9**(3) (2020)
7. Mishra, A., Jain, S.K.: A survey on question answering systems with classification. Journal of King Saud University-Computer and Information Sciences **28**(3), 345–361 (2016)
8. Dhanjal, G.S., Sharma, S., Sarao, P.K.: Gravity based punjabi question answering system. International Journal of Computer Applications **147**(3), 21 (2016)
9. Gomes Jr, J., de Mello, R.C., Str*o*ele, V., de Souza, J.F.: A hereditary attentive template-based approach for complex knowledge base question answering systems. Expert Systems with Applications **205**, 117,725 (2022)
10. Do, P., Phan, T.H.: Developing a bert based triple classification model using knowledge graph embedding for question answering system. Applied Intelligence **52**(1), 636–651 (2022)
11. Pradhan, R., Sharma, D.K.: An ensemble deep learning classifier for sentiment analysis on code-mix hindi–english data. Soft Computing pp. 1–18 (2022)
12. Oh, J.H., Torisawa, K., Hashimoto, C., Kawada, T., De Saeger, S., Wang, Y., et al.: Why question answering using sentiment analysis and word classes. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, pp. 368–378 (2012)
13. Pessutto, L., Moreira, V.: Ufrgsent at semeval-2022 task 10: Structured sentiment analysis using a question answering model. In: Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pp. 1360–1365 (2022)
14. Pradhan, R., Sharma, D.K.: A hierarchical topic modelling approach for short text clustering. International Journal of Information and Communication Technology **20**(4), 463–481 (2022)
15. da Silva, J.W.F., Venceslau, A.D.P., Sales, J.E., Maia, J.G.R., Pinheiro, V.C.M., Vidal, V.M.P.: A short survey on end-to-end simple question answering systems. Artificial Intelligence Review **53**(7), 5429–5453 (2020)
16. Gupta, D., Kumari, S., Ekbal, A., Bhattacharyya, P.: Mmqa: A multi-domain multi-lingual question-answering framework for english and hindi. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (2018)
17. Kowsher, M., Rahman, M.M., Ahmed, S.S., Prottasha, N.J.: Bangla intelligence question answering system based on mathematics and statistics. In: 2019 22nd International Conference on Computer and Information Technology (ICCIT), pp. 1–6. IEEE (2019)