IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 5, May 2024

DOI: 10.17148/IARJSET.2024.11519

NATURAL LANGUAGE PROCESSING BASED QUESTION AND ANSWER GENERATOR

D. Manoj[1], Dr. Princess maria john, Ph.D. [2]

II MCA Students, Master of Computer Applications, Hindusthan College of Engineering and Technology, Coimbatore, India[1]

Assistant Professor, Master of Computer Applications, Hindusthan College of Engineering and Technology, Coimbatore, India[2]

Abstract: In the realm of education, the creation of question papers is a fundamental yet time-consuming task for educators. With the advancements in Natural Language Processing (NLP), automated systems can now assist in generating question papers efficiently and effectively. This paper proposes a Question Paper Generator (QPG) that utilizes NLP techniques to analyze and generate questions tailored to specific educational domains. The QPG employs various NLP tasks such as text summarization, keyword extraction, and semantic analysis to understand the content of educational materials. By processing textbooks, lecture notes, and other relevant resources, the system identifies key concepts and formulates questions that assess the students' understanding of the subject matter. Additionally, the QPG ensures the questions adhere to the prescribed curriculum and learning objectives. Furthermore, the QPG incorporates features for customization, allowing educators to specify parameters such as question types (e.g., multiple choice, short answer, essay), difficulty levels, and topic preferences. Through this flexibility, the system can generate question papers that meet the diverse needs of different educational settings. Evaluation of the QPG involves comparing the generated question papers with those created manually by subject matter experts. Metrics such as accuracy, diversity, and relevance of questions are assessed to validate the effectiveness of the system. Additionally, user feedback from educators and students is gathered to refine and improve the QPG over time. The Questions are generated into two types such as subjective and objective. The application is executed using the framework called Flask. Overall, the proposed Question Paper Generator leveraging NLP techniques presents a promising solution to streamline the question paper creation process, thereby saving educators' time and ensuring the quality and relevance of assessment materials in educational settings. home.

Keywords*:* Question generator, NLTK, Natural Language Processing, POS Tagging, Flask.

I. INTRODUCTION

In the field of education, the task of creating question papers serves as a crucial element in assessing students' understanding and knowledge retention. However, this process is often labor-intensive and time-consuming for educators, requiring meticulous planning and domain expertise. With the advent of Natural Language Processing (NLP) technologies, there arises an opportunity to revolutionize the generation of question papers by leveraging computational techniques to automate and streamline this process. NLP, a subfield of artificial intelligence (AI), focuses on enabling computers to understand, interpret, and generate human language. By harnessing the power of NLP, it becomes feasible to analyze educational content, extract meaningful insights, and formulate questions that effectively evaluate students' comprehension of the subject matter. The concept of a Question Paper Generator (QPG) using NLP entails the development of an automated system capable of ingesting educational materials such as textbooks, lecture notes, and supplementary resources. Through advanced linguistic processing techniques, this system can discern key concepts, identify relevant topics, and construct questions tailored to assess students' knowledge across various domains and subjects. The integration of NLP into the question paper generation process offers several advantages. Firstly, it alleviates the burden on educators by automating the tedious task of manually creating questions. This not only saves time but also allows teachers to focus more on instructional activities and personalized student support. Secondly, NLP-powered QPGs can ensure the consistency and standardization of assessment materials, mitigating biases and discrepancies often encountered in manually curated question papers. Additionally, these systems can adapt to evolving curricula and educational standards, thereby facilitating the continuous improvement of assessment practices. Moreover, the utilization of NLP enables QPGs to incorporate adaptive features, tailoring question difficulty levels, formats, and topics to suit the

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 5, May 2024

DOI: 10.17148/IARJSET.2024.11519

diverse needs and proficiency levels of students. This adaptability fosters a more inclusive and engaging learning environment, where assessments are aligned with individual learning trajectories and educational objectives. In this context, the development and implementation of a Question Paper Generator using NLP represent a significant advancement in educational technology. By harnessing the capabilities of AI and NLP, educators can enhance the efficiency, effectiveness, and fairness of the assessment process, ultimately fostering better learning outcomes for students. This paper aims to explore the methodologies, challenges, and implications associated with the integration of NLP into question paper generation, with the ultimate goal of contributing to the advancement of educational practices in the digital age.

II. RELATED WORK

"Automated Question Paper Generation System using NLP" (Author: B. S. Bhavani, 2019): This paper presents an automated question paper generation system that utilizes NLP techniques. It discusses the process of extracting relevant keywords from textbooks, syllabi, and other educational resources using NLP, followed by the generation of questions based on these keywords. Evaluation metrics such as question relevance and diversity are considered to assess the effectiveness of the system.

"Question Generation: NLP based Approach" (Authors: Priyanka Bagdi, Ritu Tiwari, 2017): This study explores various NLP techniques for question generation in educational contexts. It discusses methods for extracting key concepts, summarizing text, and generating questions based on the extracted information. The paper also examines the challenges and limitations associated with NLP-based question generation systems and proposes potential solutions.

"Automated Question Generation using NLP Techniques" (Authors: Shraddha Agrawal, Rashmi Wadhwa, 2016): The paper presents an automated question generation system that employs NLP techniques such as part-of-speech tagging, dependency parsing, and semantic analysis. It discusses the process of analyzing text documents to identify important concepts and generate questions based on them. The study evaluates the system's performance in terms of question quality and relevance.

"An NLP-Based Approach to Generate Multiple Choice Questions from Texts" (Authors: S. Ramalingam, G. Gnanavel, 2019): This paper focuses on the generation of multiple-choice questions (MCQs) from textual content using NLP techniques. It discusses methods for extracting relevant information, formulating questions, and generating distractors for MCQs. The study evaluates the system's performance in terms of question accuracy and difficulty.

"Question Generation from Lecture Videos using NLP Techniques" (Authors: V. S. Kavitha, P. Sudha, 2020): This study explores the application of NLP techniques for generating questions from lecture videos. It discusses methods for transcribing audio content, extracting key information, and formulating questions based on the lecture content. The paper also evaluates the system's performance in terms of question coverage and coherence.

"Automated Question Generation for Educational Purposes: A Review" (Authors: Sonal Gupta, Akshita Mahajan, 2020): This review paper provides an overview of automated question generation techniques for educational purposes, with a focus on NLP-based approaches. It discusses various methods for extracting information, generating questions, and evaluating question quality. The paper also identifies research gaps and future directions in the field of automated question generation. These studies collectively highlight the potential of NLP techniques in automating the question paper generation process, thereby enhancing efficiency and effectiveness in educational assessment. However, challenges such as question diversity, quality assurance, and adaptability to different domains remain areas for further research and development.

III. METHODOLOGY

The methodology of natural language processing based question paper generator involve several steps. These steps are used to build an effective and efficient question paper generator. The first step is to input raw text data from any domain for which questions need to be generated. Preprocessing involves transforming the raw text into a more digestible form for machine learning algorithms. This includes tokenization, stemming, and lemmatization. Tokenization splits text into smaller pieces or tokens (e.g., paragraphs into sentences, sentences into words). Stemming reduces words to their root form by removing inflectional forms. Lemmatization converts words to their base form. The WordNet dataset is utilized for both multiple-choice question generation and subjective question generation. WordNet is a lexical database for the English language that provides structured semantic associations between words.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 5, May 2024

DOI: 10.17148/IARJSET.2024.11519

The NLTK library is used in Python to access WordNet and perform lemmatization. After preprocessing, the text is summarized to create multiple-choice questions. Subjective question generation involves POS tagging of the pre-processed text before summarization. Part-of-speech tagging (POS tagging) assigns the proper part of speech to each token (word) in the text, such as noun, verb, adjective, etc. This is done using tools like NLTK. Focus is placed on specific POS tags such as nouns (NN, NNS, NNP, NNPS), verbs (VB, VBN, VBD), and pronouns (PRP, PRP$). Overall, this methodology outlines the process of preprocessing raw text data, utilizing NLP techniques for feature extraction and question generation, and leveraging external datasets and libraries for enhanced functionality. This structured approach can help ensure the effectiveness and accuracy of the question and answer generation process.

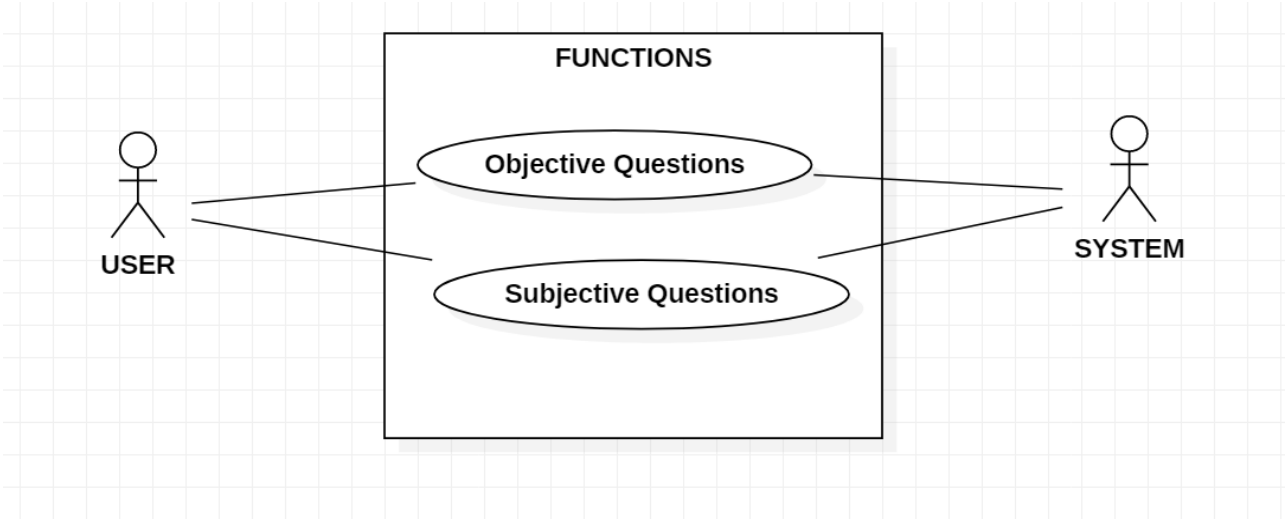

Fig. 3.1 Usecase Diagram

The steps in proposed system are:

1) Input text: The input data or text or content is given in the text box.
2) Question type: select the question type such as objective or subjective.
3) List of sentences: the given text or content is converted into list of sentences
4) List of words: the list of sentences is again converted into list of words.
5) Stop words removal: the stop words are removed from the list of words.
6) Separation of nouns: the nouns are separated from the list of words.
7) Trivial sentences: Trivial sentences are simple, straightforward statements that convey basic information without complexity or ambiguity. These trivial sentences are sorted.
8) Extract question and answer: the extracted question and answer can be printed and downloaded as CSV, EXCEL and PDF files.

IV. IMPLEMENTATION

The implementation of NLP based question paper generator is divided into three parts.

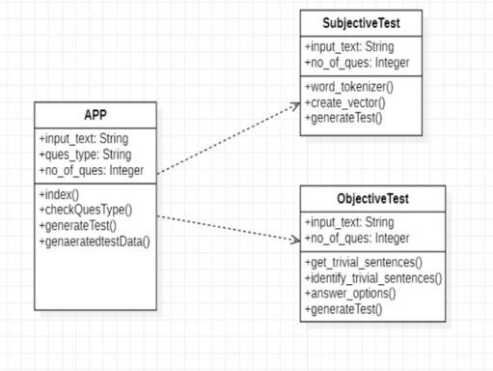

Fig. 4.1 Class Diagram

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 5, May 2024

DOI: 10.17148/IARJSET.2024.11519

4.1 Objective class

It implemented a class called `ObjectiveTest` that generates multiple-choice questions based on trivial sentences extracted from a text summary. The class takes raw data and the number of questions to be generated as input and provides methods to identify trivial sentences, generate questions, and answer options.Here's a breakdown of the key methods in the class:

1. '__init__': The constructor method initializes the `ObjectiveTest` object with the input data and the number of questions.
2. 'get_trivial_sentences': This method tokenizes the input summary into sentences and identifies trivial sentences based on specific criteria. Trivial sentences are those with simple structures or containing few words.
3. 'identify_trivial_sentences': This method further analyzes each sentence to identify noun phrases and selects key nouns to create questions. It replaces these key nouns with blanks in the sentence and generates questions accordingly.
4. 'answer_options': This method retrieves similar words (hyponyms) for a given noun using WordNet.
5. 'generate_test': This method generates a specified number of questions and corresponding answers based on trivial sentences. It selects questions with key nouns that have a length greater than the specified number of questions.

4.2 Subjective class

This implementation appears to be a class `SubjectiveTest` designed to generate subjective questions and answers based on a given text. Here's a breakdown of its main components:

1. 'Initialization': The class is initialized with the input data (`data`) and the number of questions to be generated (`noOfQues`).
2. 'Tokenization': The `word_tokenizer` method tokenizes the input text into words using NLTK's `word_tokenize` function.
3. 'Vector Creation': The `create_vector` function creates a binary vector representing the presence or absence of words in the answer tokens.
4. 'Cosine Similarity': The `cosine_similarity_score` function calculates the cosine similarity score between two vectors, representing the similarity between the answer and the input text.
5. 'Test Generation': The `generate_test` method generates subjective questions and answers. It first identifies keyword phrases in the text using a chunking grammar ('grammar'). Then, it randomly selects a keyword phrase and constructs a question using predefined question patterns ('question_pattern'). The corresponding answer is retrieved from the text. This process is repeated for the specified number of questions.
6. 'Question and Answer Lists': Finally, the method returns lists of generated questions ('que') and answers ('ans').

4.3 App class

The Flask web application that allows users to input text and select the type of test (objective or subjective) they want to generate. The application then generates test questions based on the input text and displays them to the user.
Here's a breakdown of the main components of your Flask application:

- Routes: The route renders the index.html template, which contains a form for users to input text and select the test type and number of questions.
- The test_generate route processes the form submission. It retrieves the input text, test type, and number of questions from the form. Depending on the selected test type, it generates test questions using either the ObjectiveTest or SubjectiveTest class and renders the generatedtestdata.html template to display the generated questions and answers.
- Template Rendering: The render_template function is used to render HTML templates (index.html and generatedtestdata.html) and pass data (generated questions and answers) to the templates.
- Flash Messages: Flash messages are used to display error messages if an error occurs during form submission.
- Test Generation: Depending on the selected test type (objective or subjective), the application instantiates either the ObjectiveTest or SubjectiveTest class and calls the generate_test method to generate test questions and answers based on the input text and number of questions.
- Running the Application: The if __name__ == "__main__": block ensures that the Flask application is run when the script is executed directly.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 5, May 2024

DOI: 10.17148/IARJSET.2024.11519

Sample Source Code

```
from flask import Flask, request, render_template, flash, redirect, url_for,session, Response, render_template_string
from subjective import SubjectiveTest
app.secret_key= 'aica2'
# nltk.download("all")
@app.route('/')
return render_template('index.html')
def test_generate():
inputText = request.form["itext"]
noOfQues = request.form["noq"]
objective_generator = ObjectiveTest(inputText,noOfQues)
testgenerate = zip(question_list, answer_list)

elif testType == "subjective":

question_list, answer_list = subjective_generator.generate_test()
return render_template('generatedtestdata.html', cresults = testgenerate)
flash('Error Ocuured!')
if __name__ == "__main__":
from objective import ObjectiveTest


app = Flask(__name__)
# import nltk
# exit()
def index():
@app.route('/test_generate', methods=["POST"])
if request.method == "POST":
testType = request.form["test_type"]
if testType == "objective":
question_list, answer_list = objective_generator.generate_test()
return render_template('generatedtestdata.html', cresults = testgenerate)
subjective_generator = SubjectiveTest(inputText,noOfQues)
testgenerate = zip(question_list, answer_list)

else:

return redirect(url_for('/'))
app.run(host = "0.0.0.0", port = 5001, debug=True)
```

V. RESULT ANALYSIS

An evaluation process was needed to see the success rate of a developed method. The success rate of the developed method can be seen from the achievement of the accuracy value owned, the greater the value of the accuracy the better the method developed. The process of calculating accuracy was shown in Equation 1.

Accuracy = The amount of correct data / The amount of all data

Evaluation of Automated Question Generator using Natural Language Processing (NLP) Automated question generation (AQG) systems aim to generate questions from a given text in order to help learners better understand the material. The performance of AQG systems can be evaluated in several ways, including accuracy, relevance, fluency, and diversity. Accuracy refers to the degree to which the generated questions correctly reflect the content of the text. This can be evaluated using metrics such as precision, recall, and F1-score, which measure the ability of the AQG system to generate questions that are both correct and relevant to the text. Relevance refers to the degree to which the generated questions are relevant to the content of the text and aligned with the intended learning objectives. Relevance can be evaluated by comparing the generated questions to a set of ground truth questions, or by assessing the quality of the questions based on criteria such as grammatical correctness, coherence, and topical relevance. Fluency refers to the degree to which the generated questions are grammatically correct and natural sounding. This can be evaluated by comparing the fluency of the generated questions to a set of ground truth questions, or by subjective assessments from human evaluators. Diversity refers to the degree to which the generated questions are diverse and cover a wide range of topics and perspectives. This can be evaluated by measuring the number and variety of questions generated by the AQG system, or by assessing the uniqueness of the generated questions based on criteria such as novelty and originality.

In addition to these metrics, it is also important to evaluate the usability and user experience of AQG systems. This can be done by conducting user studies to assess the effectiveness and ease of use of the AQG system, or by evaluating the user satisfaction and engagement with the generated questions. In conclusion, evaluating AQG systems requires a multi-faceted approach that considers both the technical performance of the system and the user experience. A combination of objective metrics and subjective assessments can provide a comprehensive picture of the performance of AQG systems and help to identify areas for improvement.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 5, May 2024

DOI: 10.17148/IARJSET.2024.11519

Fig 5.1 USER INTERFACE 1

Fig 5.2 USER INTERFACE 2

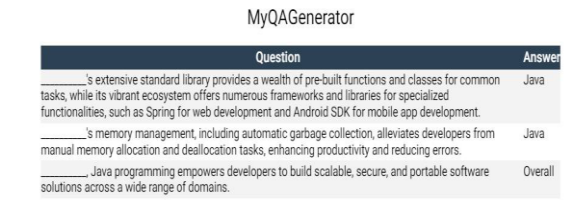

Fig 5.3 OBJECTIVE QUESTION

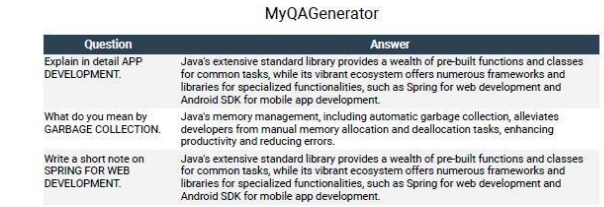

Fig 5.4 SUBJECTIVE QUESTION

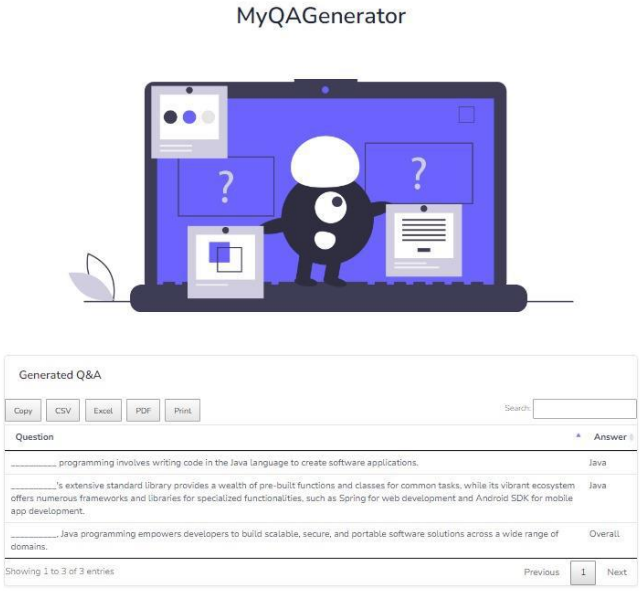

Fig 5.5 GENERATED QUESTION

VI. CONCLUSION

In conclusion, the development of a question paper generator using Natural Language Processing (NLP) offers a promising solution to streamline the process of creating diverse and relevant assessment materials in educational settings. By leveraging NLP techniques to analyze and extract information from educational texts, the generator can automate the generation of questions across various subjects and question types, including multiple-choice, fill-in-the-blank, short answer, and essay questions. The automation of question generation reduces the time and effort required for educators to create question papers, allowing them to focus more on teaching and other instructional activities. Educators can customize parameters such as topic selection, question types, and difficulty levels to tailor question papers to the specific needs and preferences of their students. By analyzing a large corpus of educational texts, the generator can produce questions that cover a wide range of topics, difficulty levels, and cognitive skills, ensuring diversity and relevance in assessment materials. The system can scale to handle large volumes of text data and adapt to changes in educational curriculum and standards, ensuring the continued relevance and currency of generated questions.

DOI: 10.17148/IARJSET.2024.11519

By providing access to high-quality assessment materials, the generator contributes to a more engaging and effective learning experience for students, helping them reinforce their understanding of course content and improve their academic performance. Through regular updates and maintenance, the generator can incorporate new data, improve NLP models, and enhance question generation algorithms, ensuring ongoing improvements in the quality and effectiveness of generated questions.

REFERENCES

[1]. Vijay Krishan Purohit', Abhijeet Kumar', Asma Jabeen' ,Saurabh Srivastava', R H Gouda' , Shinagawa, "Design of Adaptive Question Bank Development and Management System", 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, 2012

[2]. G-Asks: An Intelligent Automatic Question Generation System for Academic Writing Support by Ming Liu and Rafael A. Calv

[3]. D. R. CH and S. K. Saha , "Automatic Multiple Choice Question Generation From Text: A Survey," in IEEE Transactions on Learning Technologies, vol. 13, no. 1, pp. 14-25, 1 Jan.-March 2020, Doi: 10.1109/TLT.2018.2889100.

[4]. Anderson LW, Krathwohl DR. A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives. New York NY: Longmans; 2001.

[5]. Stephen A. Zahorian, Vishnu K. Lakdawala Oscar, R. Gonzalez, Scot Starsman, and James FLeathrum, Jr., "Question Model for Intelligent Questiong Systems in Engineering Education", 31st ASEE/IEEE Frontiers in Education Conferencee, October 10 - 13, 2001 Reno, NY, © 2001 IEEE.

[6]. Noor Hasimah Ibrahim Teo, Nordin Abu Bakar and Moamed RezduanAbd Rashid, "Representing Examination Question Knowledge into Genetic Algorithm", IEEE Global Engineering Education Conference (EDUCON), 2014.

[7]. Amruta Umardand, Ashwini – "A survey on Automatic Question Paper Generation System", International Advanced Research Journal in Science, Engineering and Technology (IARJSET), Jan 2017.

[8]. Aleena, Vidya - "Implementation of Automatic Question Paper Generator System", International Research Journal of Engineering and Technology (IRJET), Feb 2019.

[9]. Kalpana B. Khandale1, Ajitkumar Pundage, C. Namrata Mahender - "Similarities In Words Using Different Pos Taggers.", 10SR Journal of Computer Engineering (IOSR- JCE),(PP 51-55).

[10]. Edward Loper and Steven Bird "Nitk: The Natural Language Toolkit.", July 2002.

[11]. Ankita, K. A. Abdul Nazeer - "Part-Of-Speech Tagging And Named Entity Recognition Using Improved Hidden Markov Model And Bloom Filter, International Conference on Computing, Power and Communication Technologies (GUCON), 2018.

[12]. Surbhi Choudhary, Abdul Rais Abdul Waheed, Shrutika Gawandi and Kavita Joshi, "Question Paper Generator System," International Journal of Computer Science Trends and Technology, vol. 3, issue 5, Sept–Oct 2015.