AUTOMATIC QUESTION PAPER GENERATOR

A MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE

Submitted by

SANSKAR GUPTA [RA2011003011308]

PUNEET AWASTHI[RA2011003011303]

Under the guidance of

Dr Muralidharan C

Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

BONAFIDE CERTIFICATE

Certified that Mini project report titled **"AUTOMATIC QUESTION PAPER GENERATOR"** is the bona fide work of **SANSKAR AND PUNEET (RA2011003011308 AND RA2011003011303.)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

GUIDE NAME
GUIDE
Assistant Professor
Department of Computing Technologies

SIGNATURE

Dr. M. Pushpalatha
HEAD OF THE DEPARTMENT
Professor & Head
Department of Computing Technologies

ABSTRACT

An automatic question paper generator is a computer program that can create exams and tests without human intervention. The system generates a set of questions and answers based on various parameters such as subject, difficulty level, type of questions, and other criteria. The program can also analyze the previous year's question papers and generate questions based on the pattern of those papers. The generator uses natural language processing techniques to create questions that are grammatically correct and contextually relevant. The system can also incorporate different types of questions, such as multiple choice, fill-in-the-blank, and short answer questions. The benefits of an automatic question paper generator are numerous. It saves time and effort for teachers and professors who would otherwise have to create exams manually. Additionally, the system can reduce the possibility ofbias in exam creation since it is not influenced by any individual's personal beliefs or opinions. In conclusion, an automatic question paper generator is a powerful tool that can make the exam creation process faster and more efficient while maintaining objectivity and quality

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

An automatic question paper generator using AI is a sophisticated system that uses artificial intelligence and natural language processing techniques to create exams and tests without any human intervention. The system can generate questions based on various parameters, including subject, difficulty level, and question type. AI-based question paper generators have the ability to analyze a largeamount of data, including previous year's question papers, textbooks, and other reference materials, to create a comprehensive and diverse set of questions. This process ensures that the questions generated are contextually relevant and grammatically correct. Moreover, AI-based question paper generators can be customized to specific requirements and learning outcomes, ensuring that exams are aligned with the curriculum and reflect the skills and knowledge that students are expected to demonstrate. One of the key advantages of an AI-based question paper generator is that it can save time and effort for teachers and educators, who can use the generated questions as they are or modify them as per their requirements. The system can also reduce the possibility of bias in exam creation since it is not influenced by any individual's personal beliefs or opinions

CHAPTER -2

LITERATURE SURVEY

1. Automatic Question Generation from Text: This paper by Heilman et al. proposes a technique for generating questions from textual data using syntactic and semantic analysis. The proposed technique utilizes a set of predefined rules and templates to generate questions automatically.
2. A Survey of Question Generation Techniques: This paper by Ruset al. provides a comprehensive survey of various techniques used for automatic question generation. The authors categorizethe techniques based on the type of data used for question
   generation, such as text-based, image-based, and audio-based techniques.
3. Question Generation from Knowledge Graphs: This paper by Niuet al. proposes a method for generating questions from knowledge graphs. The proposed technique uses a graph-based approach to identify relevant information and generate questions automatically.
4. Automatic Generation of Multiple-choice Questions from Textbooks: This paper by Huang et al. proposes a technique for generating multiple-choice questions from textbooks using a combination of syntactic and semantic analysis. The proposed technique uses a rule-based approach to generate questions automatically.
5. A Hybrid Approach for Automatic Question Generation from Textbooks: This paper by Alamri et al. proposes a hybrid approach for generating questions from textbooks using a combination of rule-based and machine learning-based techniques.

CHAPTER-3
SYSTEM ARCHITECTURE AND DESIGN

1. **Data Collection:** The system needs to gather a large corpus ofrelevant texts and educational resources to generate questions
   from. This could include textbooks, articles, research papers, and other educational materials.
2. **Pre-processing:** Once the data is collected, the system needs topreprocess it to make it suitable for generating questions. Thiscould involve tasks such as data cleaning, text normalization, and feature extraction.
3. **Question Generation:** The question generation module is thecore component of the system. It uses AI techniques such as
   Natural Language Processing (NLP) and Machine Learning (ML) to generate questions based on the preprocessed data. The module could be trained using various algorithms like Deep Learning-based models like Transformer or GPT-3.
4. **Question Selection and Ranking:** The question generation module would produce a large number of questions, and not all of them might be relevant or suitable for the level of the students. Thus, the system would need to select the most appropriate questions based on various factors such as difficulty level, topic coverage, and relevance.
5. **Question Formatting:** Once the questions are selected, they needto be formatted into a question paper format, which could include sections like multiple-choice, short answer, and long-form questions. The system could use pre-definedtemplates or create its own formatting rules.
6. **Answer Key Generation:** Once the questions are generated, thesystem needs to generate the answer key automatically.

CHAPTER 4
METHODOLOGY

The architecture and design of an automatic question paper generator using AI can vary depending on the specific system requirements and the technologies used. Here's a general methodology that can be followed:

System Analysis: Define the system requirements, including the types of questions to be generated, difficulty levels, and other parameters. dentify the data sources, such as textbooks, previous exam papers, and other reference materials.

Natural Language Processing (NLP): Use NLP techniques such as part-of-speech tagging, entity recognition, and dependency parsing to extract relevant information from the data sources.

Machine Learning (ML): Train the ML algorithms to identify patterns and relationships between the extracted information and the parameters defined in the system requirements. The ML algorithms can be used to classify questions based on their type, difficulty level, and other parameters.

Question Generation: Generate questions using the information extracted from the data sources and the ML algorithms. The system can create different types of questions, including multiple-choice, fill- n-the-blank, and short answer questions.

Evaluation and Feedback: Evaluate the generated questions based on their relevance and correctness.

CHAPTER 5

CODING AND TESTING

```
import sqlite3from tkinter import *import webbrowserimport
smtplibimport osfrom email.mime.multipart import
MIMEMultipartfrom email.mime.text import MIMETextfrom
email.mime.base import MIMEBasefrom email import encodersimport
fpdfimport randompdf = fpdf.FPDF(format='letter')from PIL import
ImageTk, Imageconn=sqlite3.connect('paper.db')c=conn.cursor()
# to view all the table deatials in the
databasec.execute("SELECT name FROM sqlite_master where
type='table';")print(c.fetchall())
#c.execute("""CREATE TABLE EASY6(ID INT PRIMARY KEY NOT NULL, QS
CHAR(1000) NOT NULL)""")#c.execute("INSERT INTO testpaper VALUES
('Difference between A* and AO*',5,37)")#c.execute("SELECT *
FROM testpaper")#print(c.fetchall())#conn.commit()#conn.close()
def pdfgen(list1,list2,list3):
    pdf.add_page()
    pdf.set_font("Arial", size=15)
    pdf.cell(200,15,"Question paper NCU 2018", ln=1, align="C")
    pdf.set_font("Arial",'i', size=14)
    pdf.cell(200,15,"Generated using an automated paper
generation system", ln=1, align="C")
    pdf.set_font("Times", size=10)
    pdf.cell(167,15,"Max Marks : 100", align="left")
    pdf.cell(100,15,"Time : 3 Hours",ln=1, align="right")
    pdf.set_font("Arial",'b', size=13)
    pdf.cell(134,15,"Section A", align="left")
    pdf.set_font("Arial",'i', size=11)
    pdf.cell(100,15,"Max marks for this section are 4",ln=1,
align="left")
    pdf.set_font("Times", size=10)
    for i in range(5):
        pdf.cell(170,6,"Q"+str(i+1)+":
"+list1[i][0],ln=1,align="left")
    pdf.set_font("Arial",'b', size=13)
    pdf.cell(134,15,"Section B", align="left")
    pdf.set_font("Times",'i', size=11)
    pdf.cell(100,15,"Max marks for this section are 6",ln=1,
align="left")
    pdf.set_font("Arial", size=10)
    for i in range(5):
        pdf.cell(170,6,"Q"+str(i+1)+":
"+list2[i][0],ln=1,align="left")
    pdf.set_font("Arial",'b', size=13)
```

```
    pdf.cell(133,15,"Section C", align="left")
    pdf.set_font("Times",'i', size=11)
    pdf.cell(100,15,"Max marks for this section are 10",ln=1,
align="left")
    pdf.set_font("Arial", size=10)
    for i in range(5):
        pdf.cell(170,6,"Q"+str(i+1)+":
"+list3[i][0],ln=1,align="left")
    pdf.output("test.pdf")
    fromaddr = "bobbyverma96@yahoo.in"
    toaddr = "gokuverma94@gmail.com"

    msg = MIMEMultipart()

    msg['From'] = fromaddr
    msg['To'] = toaddr
    msg['Subject'] = "Generated test paper"
    body = "Here is your generated test sir"
    msg.attach(MIMEText(body, 'plain'))

    filename = "test.pdf"
    attachment = open("D:\\blah\\test.pdf", "rb")

    part = MIMEBase('application', 'octet-stream')
    part.set_payload((attachment).read())
    encoders.encode_base64(part)
    part.add_header('Content-Disposition', "attachment;
filename= %s" % filename)

    msg.attach(part)
    conn=smtplib.SMTP('smtp.gmail.com',587)
    conn.ehlo()
    conn.starttls()
    conn.login('gokuverma94@gmail.com',os.environ["pass"])
    text = msg.as_string()
    conn.sendmail(fromaddr, toaddr, text)
    conn.quit()
def addqs(qs, marks, diff):
    c.execute("SELECT MAX(ID) FROM "+diff+marks)
    data = c.fetchone()
    if data[0] is None:
        i = 1
    else:
        i = data[0] + 1
    c.execute("INSERT INTO "+diff+marks+"
VALUES("+str(i)+",\""+qs+"\");")
    print(qs)
```

```
    conn.commit()

def ques_select(diff3):
    c.execute("SELECT MAX(ID) FROM "+diff3+"4")
    data1=c.fetchone()
    c.execute("SELECT MAX(ID) FROM "+diff3+"6")
    data2=c.fetchone()
    c.execute("SELECT MAX(ID) FROM "+diff3+"10")
    data3=c.fetchone()

    if data1[0]<5 or data2[0]<5 or data3[0]<5:
        print("Not sufficient elements in Tables")
        exit()
    else:
        rand1 = random_num_gen(data1[0])
        rand2 = random_num_gen(data2[0])
        rand3 = random_num_gen(data3[0])
    obj1 = []
    obj2 = []
    obj3 = []
    for i in range(5):
        c.execute("SELECT QS FROM "+diff3+"4 WHERE "+"ID =
"+str(rand1[i]))
        obj1.append(list(c.fetchone()))
        c.execute("SELECT QS FROM "+diff3+"6 WHERE "+"ID =
"+str(rand2[i]))
        obj2.append(list(c.fetchone()))
        c.execute("SELECT QS FROM "+diff3+"10 WHERE "+"ID =
"+str(rand3[i]))
        obj3.append(list(c.fetchone()))
    pdfgen(obj1,obj2,obj3)
    def random_num_gen(n):
    rlist = random.sample(range(n),5)
    rlist = [x+1 for x in rlist]
    return rlist
def genwin():
    main=Toplevel()
    main.geometry("600x400+8+400")
    main.title("Generate Test")

canvas=Canvas(main,width=470,height=80,relief='raised',borderwid
th=3)
    canvas.grid(row=0,column=1,padx=50,pady=20)
    canvas.create_text(250,50,fill="blue",font="Times 13 italic
bold",
                           text="Generated test will be e-
mailed to you!!!")
```

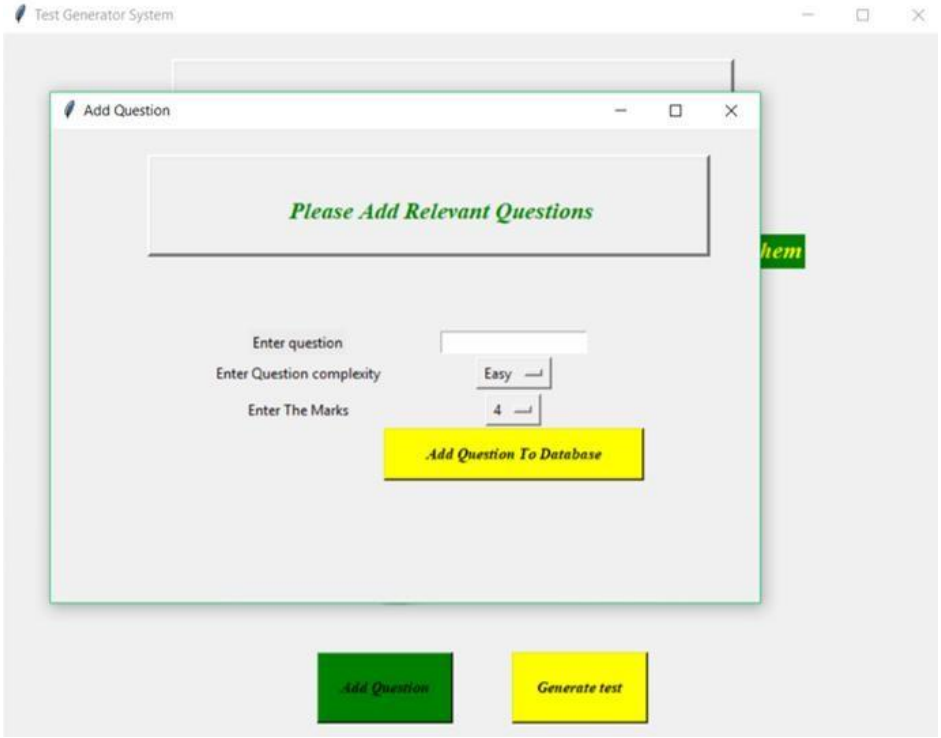
```
    def click2():
        diff3=variable.get()
        ques_select(diff3)

    frame=Frame(main)
    frame.grid(row=1,column=1,padx=50,pady=40)
    frame2=Frame(frame)
    frame2.grid(row=0,column=1,padx=50,pady=40)
    button1=Button(frame,text="Generate test",font="Times 10
italic bold")
    gg=Label(frame,text="Please select the complexity level")
    gg.grid(row=0,column=0)
    OPTIONS = ['Easy','Medium','Hard']
    variable = StringVar()
    variable.set(OPTIONS[0])

    w = OptionMenu(frame2, variable, *OPTIONS)
    w.grid(row=0,column=1)
    button1.config( height =2, width = 13,bg="yellow",
command=click2)
    button1.grid(row=1,column=0,padx=10,pady=30)
def addwin():
    main=Toplevel()
    main.geometry("600x400+8+400")
    main.title("Add Question")

canvas=Canvas(main,width=470,height=80,relief='raised',borderwid
th=3)
    canvas.grid(row=0,column=1,padx=80,pady=20)
    canvas.create_text(250,50,fill="green",font="Times 15 italic
bold",
                           text="Please Add Relevant Questions")
    def click1():
        ques=namez.get()
        diff=variable1.get()
        marks=variable2.get()
        c.execute("create table if not exists "+diff+marks+" (ID
INT PRIMARY KEY NOT NULL, QS CHAR(1000) NOT NULL)")
        addqs(ques, marks, diff)
    frame=Frame(main)
    lbl1 = Label(frame, text = "Enter
question",background='#ECECEC')
    lbl1.grid(row=0,column=0,pady=0)
    namez=Entry(frame)
    namez.grid(row=0,column=1,pady=0)
    lbl2=Label(frame,text="Enter Question complexity")
```

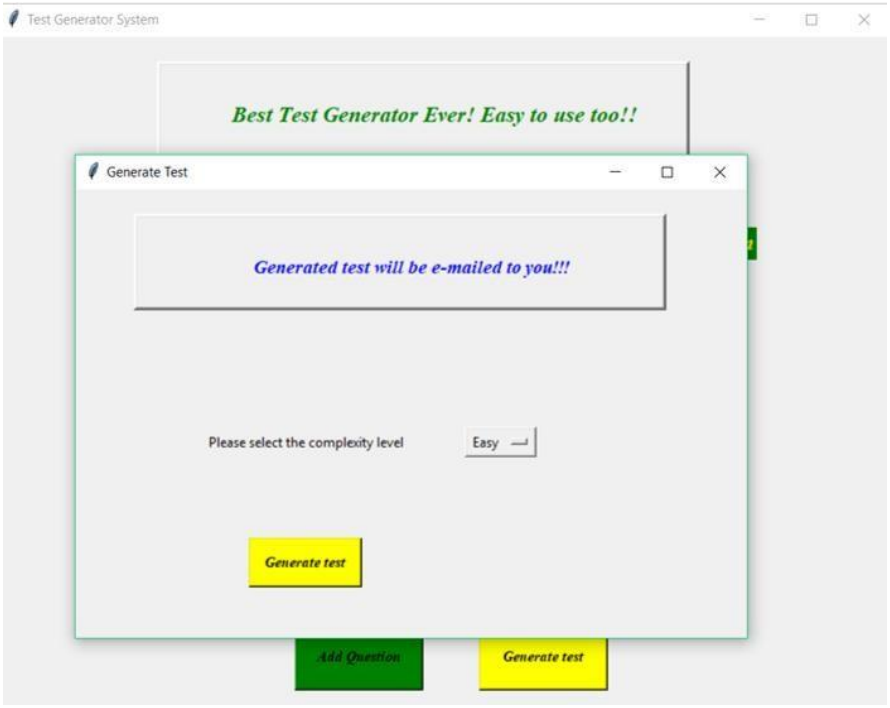
```
    lbl2.grid(row=1,column=0)
    OPTIONS = ['Easy','Medium','Hard']
    variable1 = StringVar()
    variable1.set(OPTIONS[0])
    namez2 = OptionMenu(frame, variable1, *OPTIONS)
    namez2.grid(row=1,column=1)

    lbl3=Label(frame,text="Enter The Marks")
    lbl3.grid(row=2,column=0)
    OPTIONS = [4,6,10]
    variable2 = StringVar()
    variable2.set(OPTIONS[0])
    namez3 = OptionMenu(frame, variable2, *OPTIONS)
    namez3.grid(row=2,column=1)
    frame.grid(row=1,column=1,padx=50,pady=40)
    button1=Button(frame,text="Add Question To
Database",font="Times 10 italic bold",command=click1)
    button1.config( height =2, width = 30,bg="yellow")
    button1.grid(row=3,column=1,padx=0)
def mainwin():
    main = Tk()
    main.title("Test Generator System")
    img = ImageTk.PhotoImage(Image.open("a.gif"))
    main.geometry("800x600+8+400")


canvas=Canvas(main,width=470,height=80,relief='raised',borderwid
th=3)
    canvas.grid(row=0,column=4,padx=80,pady=20)
    canvas.create_text(250,50,fill="green",font="Times 15 italic
bold",
                            text="Best Test Generator Ever! Easy
to use too!!")

    text=Label(main,text="Welcome to Test Generator, if you have
new questions please add them",font="Times 15 italic bold")
    text.configure(bg="green",foreground="yellow")
    text.grid(row=1,column=4,padx=80,pady=40)
    panel = Label(main, image = img,height=220,width=600)
    panel.grid(row=3,column=4,padx=80,pady=20)
    frame=Frame(main)
    frame.grid(row=4,column=4,padx=50,pady=20)
    button1=Button(frame,text="Add Question",font="Times 10
italic bold",command=addwin)
    button1.config( height =3, width = 15,bg="green")
    button1.grid(row=0,column=0,padx=50)
```

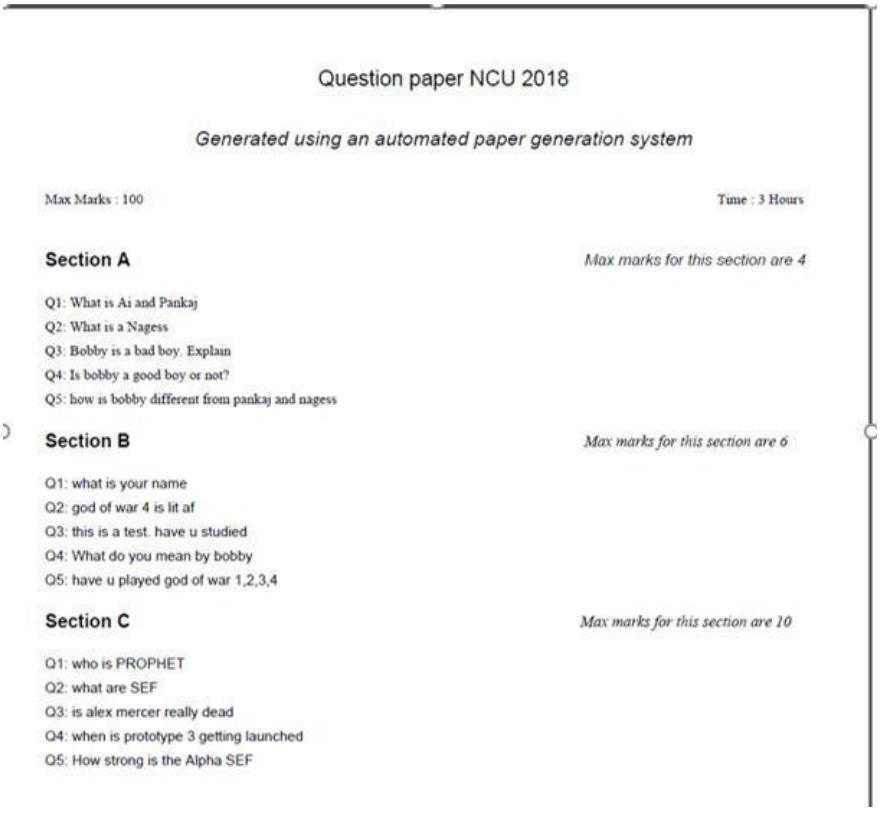
```
    button2=Button(frame,text="Generate test",font="Times 10
italic bold",command=genwin)
    button2.config( height =3, width = 15,bg="yellow")
    button2.grid(row=0,column=1)
    main.mainloop()mainwin()
```

CHAPTER 6

SCREENSHOTS AND RESULTS







Question paper NCU 2018

Generated using an automated paper generation system

Max Marks : 100 Time : 3 Hours

Section A *Max marks for this section are 4*

Q1: What is Ai and Pankaj
Q2: What is a Nagess
Q3: Bobby is a bad boy. Explain
Q4: Is bobby a good boy or not?
Q5: how is bobby different from pankaj and nagess

Section B *Max marks for this section are 6*

Q1: what is your name
Q2: god of war 4 is lit af
Q3: this is a test. have u studied
Q4: What do you mean by bobby
Q5: have u played god of war 1,2,3,4

Section C *Max marks for this section are 10*

Q1: who is PROPHET
Q2: what are SEF
Q3: is alex mercer really dead
Q4: when is prototype 3 getting launched
Q5: How strong is the Alpha SEF

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENTS

In conclusion, the automatic question paper generator using AI is a powerful tool that can revolutionize the process of creating exams and tests. The system can generate questions
based on various parameters such as subject, difficulty level, and question type. It uses
natural language processing techniques and machine learning algorithms to create
contextually relevant and grammatically correct questions.

The benefits of using an automatic question paper generator using AI are numerous, including saving time and effort for educators and institutions, reducing the possibility of bias in exam creation, and improving the efficiency of the exam creation process.

In the future, there are several enhancements that can be made to the automatic question paper generator using AI. For instance, the system can be improved to generate questions that are more complex and require higher-order thinking skills. Additionally, the generator can be integrated with learning management systems and other educational technologies to provide a comprehensive platform for teaching and assessment.

Moreover, the system can be enhanced to provide personalized feedback to students based on their performance on exams. This can help students identify their strengths and weaknesses and improve their learning outcomes.

Overall, the automatic question paper generator using AI has the potential to transform the way exams and tests are created, and with continued research and development,will become an essential tool for educators and institutions in the future

REFERENCES

Here are some additional references for automatic question paper generator using AI:

"Automated Question Paper Generation System Using Machine Learning" by A. K. Mishra, M. M. Goswami, and R. K. Tiwari. International Journal of Computer Applications, Vol. 179, No. 20, 2018.

"An Automatic Question Paper Generator System using Natural Language Processing Techniques" by A. R. Kumar, R. Aravindan, and S. U. N. Raja. International Journal of Pure and Applied Mathematics, Vol. 118, No. 20, 2018.

"Question Paper Generator for Online Examination using Deep Learning" by K. R. Gadekar and P. V. Ingole. International Journal of Advanced Research in Computer Science, Vol. 9, No. 3, 2018.

"A Hybrid Technique for Automatic Question Paper Generation using NLP and Ontology" by S. B. Pande and S. M. Ghosh. International Journal of Computer Science and Information Technologies, Vol. 6, No. 5, 2015.

"An Intelligent Question Paper Generation System using Deep Learning and Natural Language Processing" by R. K. Gupta and S. R.

Chakraborty. International Journal of Innovative Research in Computer and Communication Engineering, Vol. 7, No. 6, 2019.

These papers provide additional insights into the design, implementation, and evaluation of automatic question paper generator using AI