assignment-akaike-nlp

October 24, 2023

1 MCQ Question Generator using Spacy Library

This code generates multiple-choice questions (MCQs) based on a given context paragraph using the Spacy library. The MCQs are designed to have multiple correct answer choices for added variety. The generated MCQs are then displayed to the user.

2 Import necessary libraries

Imports the required libraries, spacy for natural language processing and random for generating random choices.

[1]:
```
import spacy
import random
```

[2]:
```
spacy.cli.download("en_core_web_sm")
nlp = spacy.load("en_core_web_sm")
```

```
 Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
```

3 Define function

Defines a function get_mca_questions that takes a context paragraph and the number of questions to generate.

[3]:
```
def get_mca_questions(context: str, num_questions: int):
  doc = nlp(context)
```

4 Define MCQ generation function

Defines a function generate_mcq_with_multiple_correct to create MCQs with multiple correct answers.

[4]:
```
def generate_mcq_with_multiple_correct(question, correct_answers,␣
↪other_options, num_options=4):
        options = correct_answers + other_options
        random.shuffle(options)
```

```
        mcq = {
            "question": question,
            "options": options,
            "correct_answers": correct_answers
        }
```

5 Generate a variety question

Defines a function generate_variety_question that randomly selects a sentence and a word from the context to create a fill-in-the-blank question

[5]:
```
def generate_variety_question():
        # randomly select the sentence from content
        sentence = random.choice(list(doc.sents))

        # randomly choose non- pronounciation words from sentence as blank word
        blank_word = random.choice([token for token in sentence if not token.
↪is_punct])

        # create a question text with blank word ----
        question_text = sentence.text.replace(blank_word.text, "______")

        #set correct answers to the blank word
        correct_answers = [blank_word.text]

        #generating other possible answers
        other_options = [token.text for token in doc if token.is_alpha and␣
↪token.text != correct_answers[0]]

        #randonly determine how many correct options
        num_correct_options = random.randint(1, 2)

        #randomly select correct options to the list of options
        correct_answers.extend(random.sample(other_options,␣
↪num_correct_options))

        # no of incorrect answers
        num_other_options = min(4 - num_correct_options, len(other_options))
        other_options = random.sample(other_options, num_other_options)

        #generationg final MCQ
        mcq = generate_mcq_with_multiple_correct(question_text,␣
↪correct_answers, other_options)
        return mcq
```

6 Generate questions & Process and format questions

Generates a list of questions using the generate_variety_question function based on the context and specified number of questions. Formats the generated questions, options, and correct answers into strings.

[ ]:
```
questions = [generate_variety_question() for _ in range(num_questions)]
#created empty list to store multiple choice questions
    mca_questions = []

 # enumerate function is used to iterate over the questions
    for i, question in enumerate(questions, start=1):

        #created a string for question number and question text.
        question_str = f"Q{i}: {question['question']}\n"

        #created empty string to store option for current question
        options_str = ""

        #iterate through options
        for j, option in enumerate(question['options']):
            options_str += f"{j+1}. {option}\n"

        #format the correct answers into human redable format
        correct_options_formatted = " & ".join([f"({chr(97+question['options'].
↪index(ans))})" for ans in question['correct_answers']])

        #combine the questions and options and format the correct answes
        correct_options_str = f"Correct Options: {correct_options_formatted}"

        #add the questions into formated questions
        mca_question = f"{question_str}{options_str}{correct_options_str}\n"
        mca_questions.append(mca_question)

    #return the MCQ questions
    return mca_questions
```

7 Print Questions

Takes user input for the context and number of questions, generates MCQs, and prints each question along with options and correct answers.

[ ]:
```
context = input("Enter the paragraph: ")
num_questions = int(input("Enter the number of questions: "))
mca_questions = get_mca_questions(context, num_questions)
for question in mca_questions:
    print(question)
```

8 final code

[1]:
```
import spacy
import random

# Load English language model
nlp = spacy.load("en_core_web_sm")

def get_mca_questions(context: str, num_questions: int):
    doc = nlp(context)

    def generate_mcq_with_multiple_correct(question, correct_answers,␣
 ↪other_options, num_options=4):
        options = correct_answers + other_options
        random.shuffle(options)

        mcq = {
            "question": question,
            "options": options,
            "correct_answers": correct_answers
        }

        return mcq

    def generate_variety_question():
        sentence = random.choice(list(doc.sents))
        blank_word = random.choice([token for token in sentence if not token.
 ↪is_punct])

        question_text = sentence.text.replace(blank_word.text, "_____")
        correct_answers = [blank_word.text]

        other_options = [token.text for token in doc if token.is_alpha and␣
 ↪token.text != correct_answers[0]]
        num_correct_options = random.randint(1, 2)  # Generate 1 or 2 correct␣
 ↪options
        correct_answers.extend(random.sample(other_options,␣
 ↪num_correct_options))

        num_other_options = min(4 - num_correct_options, len(other_options))
        other_options = random.sample(other_options, num_other_options)

        mcq = generate_mcq_with_multiple_correct(question_text,␣
 ↪correct_answers, other_options)
        return mcq

    questions = [generate_variety_question() for _ in range(num_questions)]
```

```
    mca_questions = []
    for i, question in enumerate(questions, start=1):
        question_str = f"Q{i}: {question['question']}\n"
        options_str = ""
        for j, option in enumerate(question['options']):
            options_str += f"{j+1}. {option}\n"

        correct_options_formatted = " & ".join([f"({chr(97+question['options'].
↪index(ans))})" for ans in question['correct_answers']])
        correct_options_str = f"Correct Options: {correct_options_formatted}"

        mca_question = f"{question_str}{options_str}{correct_options_str}\n"
        mca_questions.append(mca_question)

    return mca_questions

context = input("Enter the paragraph: ")
num_questions = int(input("Enter the number of questions: "))
mca_questions = get_mca_questions(context, num_questions)
for question in mca_questions:
    print(question)
```

```
Enter the paragraph:  So another crushing win for South Africa at the Wankhede.
The ball just seems to do a bit more under lights over here, and their bowlers
exploited it beautifully once again as they did against England. Marco Jansen
was superb again in his opening spell, troubling batters with pace, movement and
extra bounce. Lizaad Williams, coming in for Ngidi, also found movement. They
had Bangladesh at 31/3 inside 8 overs, and then Rabada and Coetzee took over the
mantle of wicket-taking as they generally do. Even in the 15th over, the ball
was nipping around, as evidenced by a beauty from Rabada to trap Litton Das LBW.
Of course, the match was over by then. Bangladesh did have something to cheer
about towards the end as Mahmudullah played beautifully to smash a hundred while
the tail wagged in support. Still, they fell short by 149 runs - a reflection of
the challenge they were facing after being hammered by de Kock and Klaasen
earlier. South Africa smashed 143 in the last 10 overs against England, and 144
in the last 10 overs today. Bangladesh weren't coming back from that.
Enter the number of questions: 5
Q1: Bangladesh did have something to cheer about towards the end ______
Mahmudullah played beautifully to sm______h a hundred while the tail wagged in
support.
1. bounce
2. to
3. their
4. while
5. as
Correct Options: (e) & (c) & (d)
```

```
Q2: Bangladesh were______ coming back from that.
1. in
2. n't
3. Das
4. mantle
5. were
Correct Options: (b) & (a) & (e)

Q3: Even in the 15th over, the b______ll w______s nipping ______round, ______s
evidenced by ______ be______uty from R______b______d______ to tr______p Litton
D______s LBW.
1. to
2. it
3. South
4. a
5. Bangladesh
Correct Options: (d) & (c)

Q4: Marco ______ was superb again in his opening spell, troubling batters with
pace, movement and extra bounce.
1. to
2. South
3. by
4. then
5. Jansen
Correct Options: (e) & (b)

Q5: South Africa ______ 143 in the last 10 overs against England, and 144 in the
last 10 overs today.
1. inside
2. match
3. they
4. smashed
5. in
Correct Options: (d) & (a)
```

[ ]: