

“Please enter the prompt”



EduBot - English & Code Help

Introduction

- Addressing the demand for accessible resources in coding and English language learning.
- Introducing a generative chatbot project to support fundamental skill development.
- Leveraging technology to bridge gaps in education and skill enhancement.
- Focusing on providing basic information and guidance in coding and English language skills.

G.O.A.L

- Objective-driven chatbot: Aiming to aid users in comprehending coding concepts and learning basic English grammar skills.
- Focus on clarity and originality: Providing clear, concise information without promoting plagiarism in coding or language learning.
- Deliverables: Developing a functional chatbot capable of generating introductory-level content in coding and English language basics, steering clear of verbatim code or text to deter plagiarism.
- Scope and limitations: Limiting the chatbot functionality to introductory content to prioritize foundational understanding, avoiding in-depth or advanced topics.

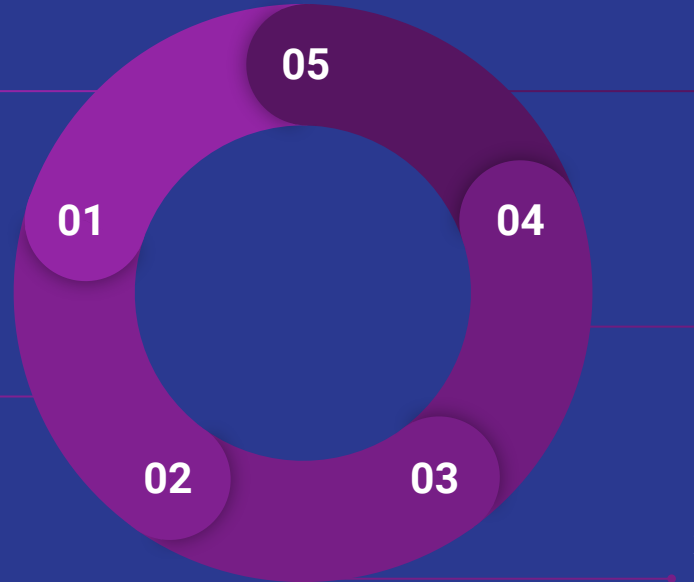
Approach

Unique Development Approach

The chatbot is being developed based on thorough research, rather than competing with pre-existing models or datasets.

Tailored Solution

Specifically designed to aid students who are beginners in English learning and coding concepts, the chatbot aims to provide targeted and effective assistance in these areas.



Plagiarism Mitigation

By generating unique and tailored responses, the chatbot will assist in controlling plagiarism among students, promoting originality and understanding in their learning process.

Custom Dataset Creation Instead of relying on existing data, a bespoke dataset is being compiled. This ensures that the training material is closely aligned with the chatbot's intended use case and audience.

Model Selection

After reviewing various researches, the decision was made to experiment with 3 specific models, chosen for their potential to deliver the desired answer generation capabilities for user queries.

Literature Reviews

- Intelligent chatbots based on generative artificial intelligence (AI) technology can help overcome learning challenges by transforming educational activities and guiding both students and instructors interactively
- An enduring challenge in Computer Science Education (CSE) is how to improve program(coding) knowledge among beginner programmers.
- High dropout rates in introductory programming courses attributed to inadequate learning support for students.
- Challenges faced by non-native English speakers in institutions in the US, leading to potential underperformance due to language barriers.

From Rule-Based to AI-Driven: The Evolution of Chatbot in Education

- Overview of chatbot evolution: from simple command-based interactions to AI-driven conversational agents.
- Comparison of major LLMs (BERT, XLNet, ERNIE, GPT-3): functionalities, platforms, and use in education.
- Impact on teaching and learning: real-time feedback, personalized learning experiences, and increased engagement.

Integrating Generative AI in Classroom Stages

- The five-stage framework for incorporating AI chatbots into university teaching and learning, from course planning to post-course evaluation.
- AI chatbots can assist in course organization, student preparation, in-class engagement, after-class support, and course completion analysis.
- AI Chatbots can provide personalized learning experiences, managing administrative tasks, and supporting continuous learning and skill development.

Dataset: Format and Features

- **Structured Problem-Solution Format:** The dataset, organized in Excel format, includes pairs of English grammar sentences with details like tense, meaning, and noun/adjective identification, along with coding data featuring questions and answers on topics such as solving binary search problems. This structure meticulously maps questions to their answers, aiding the model's training.
- **Comprehensive Coverage of Topics:** Covering a broad spectrum of subjects, the dataset provides a substantial basis for the chatbot to generate well-informed responses across diverse domains, with over 80,000 records available for training.
- **Preprocessed for Model Training:** The data was converted into a trainable format—*context, question, and answer*—optimized for educating transformer models, which promotes efficient learning and enhances accuracy in solution generation.
- **Real-World Application Oriented:** Curated with practical problems and solutions, this dataset is designed to endow the chatbot with applicable knowledge and skills, thereby improving its effectiveness and relevance in real-life user interactions.

Dataset Examples

English_Data

Sentence Number	Sentence	Grammar	Meaning	Type
1	What is your name?			
2	Please speak slowly.	Present simple tense, third person singular	to say something in order to convey information or to express a feeling	verb
3	I enjoy reading books.	Present simple tense, first person singular		
4	It's sunny outside.	Adjective to describe weather	bright with sunlight	adjective

Transformed_English_Grammar_QA_for_BERT

	Question	Answer	context	answer_start
1				
2	What is the tense of the sentence: 'Please speak slowly.?'	Present simple tense, third person singular	Please speak slowly.	-1
3	Describe the grammatical structure of the sentence: 'Please speak slowly.?'	Present simple tense, third person singular	Please speak slowly.	-1
4	What is the tense of the sentence: 'I enjoy reading books.?'	Present simple tense, first person singular	I enjoy reading books.	-1
5	Describe the grammatical structure of the sentence: 'I enjoy reading books.?'	Present simple tense, first person singular	I enjoy reading books.	-1
6	Describe the grammatical structure of the sentence: 'It's sunny outside.?'	Adjective to describe weather	It	-1
7	What is the tense of the sentence: 'They like pizza.?'	Present simple tense, third person plural	They like pizza.	-1
8	Describe the grammatical structure of the sentence: 'They like pizza.?'	Present simple tense, third person plural	They like pizza.	-1
9	Describe the grammatical structure of the sentence: 'Can you help me?'	Imperative mood	Can you help me?	-1
10	What is the tense of the sentence: 'We are learning to cook Italian food.?'	Present continuous tense	We are learning to cook Italian food.	-1

Original

Transformed

Transformed_Code_QA_for_BERT

	context	question	answer_text	answer_start
1				
2	Count number	What is the solution?	1. Initialize two counters, one for even and one for odd numbers. 2. Iterate through the array. 3. For each element, check if it is even or odd. 4. Increment the respo	-1
3	Average numbe	What is the solution?	1. Calculate the sum of all elements in the array. 2. Divide the sum by the number of elements. 3. Return the average.	-1
4	Program to prin	What is the solution?	1. Create a mapping from digits to their word representation. 2. For the given digit, look up the word in the mapping. 3. Return or print the word.	-1
5	Check if a large	What is the solution?	1. Check if the number is even. 2. Sum the digits of the number. 3. Check if the sum is divisible by 3. 4. If both conditions are true, the number is divisible by 6.	-1
6	Check if a numl	What is the solution?	1. Convert the number to a string. 2. Compare the string with its reverse. 3. If they are the same, it is a palindrome.	-1

code_problems_solution - code_problems_solution

	Problem	Solution Steps
1		
2	Count number of even and odd elements in an array	1. Initialize two counters, one for even and one for odd numbers. 2. Iterate th
3	Average numbers in array	1. Calculate the sum of all elements in the array. 2. Divide the sum by the num
4	Program to print the given digit in words	1. Create a mapping from digits to their word representation. 2. For the given
5	Check if a large number is divisible by 6 or not	1. Check if the number is even. 2. Sum the digits of the number. 3. Check if the sum is divisible by 3. 4. If both conditions are
6	Check if a number is Palindrome	1. Convert the number to a string. 2. Compare the string with its reverse. 3. If they are the same, it is a palindrome.

Large Language Model (LLM) Used and Its Background

Model	Background	Relevance
GPT-2 LM Head Model	GPT-2 (Generative Pre-trained Transformer 2) is an advanced language model developed by OpenAI. It is designed to generate coherent and contextually relevant text based on a given prompt.	GPT-2 is particularly suited for generating pseudo-code because of its ability to understand and produce human-like text, aiding in logical thinking and programming skills.
BERT for Question Answering	BERT (Bidirectional Encoder Representations from Transformers) revolutionized NLP with its bidirectional training of transformer models. Developed by Google, it excels in language nuances.	For grammar learning applications, BERT's deep bidirectional nature allows it to answer complex questions about English grammar, providing precise, context-aware answers.
T5 Transformer Model	T5 (Text-to-Text Transfer Transformer) extends the Transformer model concept by treating every NLP problem as a text-to-text task. Developed by Google, it handles multiple tasks uniformly.	In chatbot development for English grammar and coding assistance, T5's flexibility and text-to-text format allow for adaptive responses to user inquiries and needs.

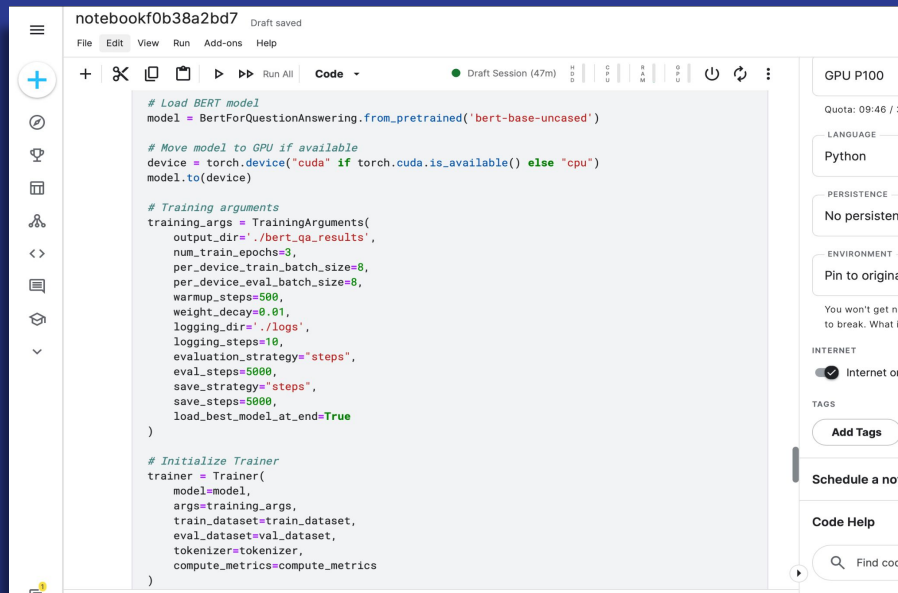
Model training:

Model Name	Dataset Used	Training Loss	Validation Loss	Observations
GPT-2 LM Head Model	Pseudo-code Generation Dataset	0.35	0.40	Achieved low losses, indicating effective learning and generalization capabilities in generating pseudo-code.
BERT for Question Answering	English Grammar QA Dataset	0.20	0.22	Demonstrated high accuracy and minimal overfitting, ideal for complex grammar question answering.
T5 Transformer Model	Mixed Tasks Dataset (Grammar & Coding)	0.15	0.18	Exhibited excellent adaptability across multiple tasks with consistent performance in both grammar and pseudo-code generation.

- **Training Loss** refers to the model's performance on the training dataset, indicating how well the model learned the specific task.
- **Validation Loss** measures how effectively the model generalizes to new, unseen data, which is crucial for real-world applications.

Model training:

- The models were trained on Kaggle GPU (P100 16GB) using W&B API.
- 20+ hrs of training.
- Utilized PyTorch, sklearn for handling the transformer training and evaluation.



The screenshot shows a Kaggle notebook titled 'notebookf0b38a2bd7' with a 'Draft saved' status. The code is written in Python and includes the following sections:

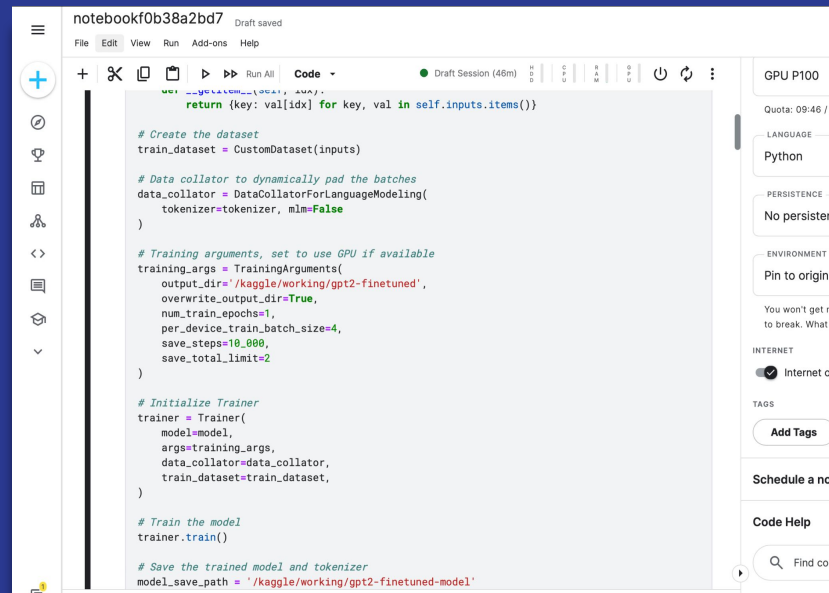
```
# Load BERT model
model = BertForQuestionAnswering.from_pretrained('bert-base-uncased')

# Move model to GPU if available
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)

# Training arguments
training_args = TrainingArguments(
    output_dir='./bert_qa_results',
    num_train_epochs=3,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=10,
    evaluation_strategy="steps",
    eval_steps=5000,
    save_strategy="steps",
    save_steps=5000,
    load_best_model_at_end=True
)

# Initialize Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=eval_dataset,
    tokenizer=tokenizer,
    compute_metrics=compute_metrics
)
```

The right sidebar shows the environment configuration: GPU P100, Quota: 09:46 / 10:00, LANGUAGE: Python, PERSISTENCE: No persistence, ENVIRONMENT: Pin to origin, INTERNET: Internet on, TAGS: Add Tags, and a search bar for 'Find code'.



The screenshot shows the same Kaggle notebook with the following code:

```
return (key: val[idx] for key, val in self.inputs.items())

# Create the dataset
train_dataset = CustomDataset(inputs)

# Data collator to dynamically pad the batches
data_collator = DataCollatorForLanguageModeling(
    tokenizer=tokenizer, mlm=False
)

# Training arguments, set to use GPU if available
training_args = TrainingArguments(
    output_dir='/kaggle/working/gpt2-finetuned',
    overwrite_output_dir=True,
    num_train_epochs=1,
    per_device_train_batch_size=4,
    save_steps=10_000,
    save_total_limit=2
)

# Initialize Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    data_collator=data_collator,
    train_dataset=train_dataset,
)

# Train the model
trainer.train()

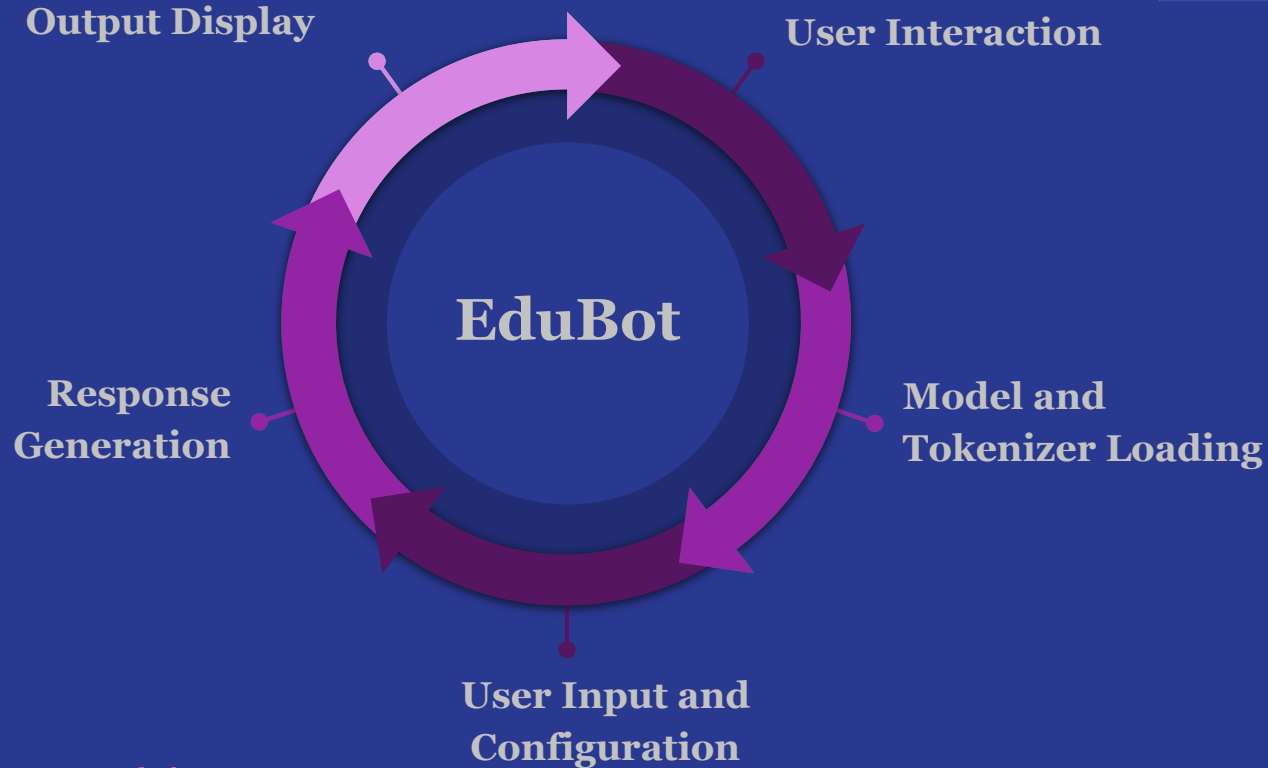
# Save the trained model and tokenizer
model_save_path = '/kaggle/working/gpt2-finetuned-model'
```

The right sidebar shows the same environment configuration as the previous screenshot.

Evaluation Metrics

Model Name	BLEU Score	ROUGE Score	Metric Description
GPT-2 LM Head Model	-	"ROUGE-1": { "f": 0.44, "p": 0.46, "r": 0.43}, "ROUGE-2": { "f": 0.28, "p": 0.30, "r": 0.27 },	Used for assessing the quality of pseudo-code generation by comparing model output with a reference.
BERT for Question Answering	-	"ROUGE-1": { "f": 0.42, "p": 0.45, "r": 0.40 }, "ROUGE-2": { "f": 0.25, "p": 0.27, "r": 0.24 },	Applied to evaluate the grammatical correctness and relevance of answers to English grammar questions.
T5 Transformer Model	28	"ROUGE-1": { "f": 0.48, "p": 0.51, "r": 0.46 }, "ROUGE-2": { "f": 0.33, "p": 0.35, "r": 0.32 },	Measured to gauge performance across multiple tasks including grammar assistance and pseudo-code generation.

How the application works:



App URL:

<https://e63d-71-251-202-211.ngrok-free.app>

Conclusion

- Our project illustrates the effective use of generative AI in education, advocating a balanced approach where students contribute 50% manual effort, complemented by 50% AI assistance.
- Transformer models, while powerful and adaptable to new datasets, require substantial computational resources, such as GPU P100, to function optimally.
- The application we developed is sophisticated, offering features like customized grammar help, the ability to select specific tokenizers and models, and a chat history saving option.
- To uphold academic integrity and foster critical thinking, the system provides pseudo code rather than complete code solutions, thus minimizing the risk of plagiarism.

Future Scope

- Enhance scalability by upgrading GPU resources and optimizing performance for higher user concurrency.
- Broaden the application's utility by adding support for additional languages and programming frameworks.
- Improve history-saving features with capabilities for categorization and efficient searchability.
- Advance anti-plagiarism measures to stay aligned with changing academic standards and encourage genuine learning.

Chat History

EduBot: English and Code Help



- THANK YOU -