|  |  |
| --- | --- |
| **Level 5 AI and Machine Learning Bootcamp – Project Plan** | |
|  | |
| **Student:** | **Spencer David O’Hegarty** |
| **Assessor:** | **John McKechnie** |
| **Hand in Deadline:** | **11th April 2025** |

**Project Objectives**

The use of AI in education has the potential to support pupils learning and help reduce teacher workload. The Department for Education (DfE) continues to commit heavily in the pursuit of artificial intelligence (AI) technologies to reduce teacher workload and improve student experience. Moreover, the DfE is embracing AI to enhance STEM education and streamline teaching practices. Generative AI tools are being integrated to reduce administrative burdens, like lesson planning and marking, enabling teachers to focus on student engagement. In the spirit of this endeavour, this project aims to create a prototype model to assist students in mathematical reasoning through use of comprehension math problems whereby students are asked tailored math questions and are offered a correct answer response. The aim is not to a create a fully interactive, production ready system, rather the intention is to produce the basis of a model capable of answering bespoke comprehension maths questions with a view to further development beyond the scope of this project. The core project objectives can be summarised as follows,

* Develop a mechanism by which to automatically generate a dataset of comprehension maths questions;
* Develop an appropriate model which is trained to respond correctly to math comprehension questions;
* Employ relevant evaluation techniques to measure model performance;
* Iteratively improve the model based on evaluation feedback;
* Provide reasonable and justifiable conclusions concerning the performance of the final model;
* Provide recommendations for further development based on lessons learnt.

**The Github repo for this project can be found here:** <https://github.com/spanersoraferty/math-problem-assistant>

**Data Considerations**

When building an **encoder-decoder sequence-to-sequence model** for our math comprehension assistant, the following key factors will be considered,

**1. Data Quality & Diversity**

* The dataset should include a variety of comprehension math problems, with a mix of narrative and arithmetic.
* The data should ensure a balanced representation of different problem types to prevent bias toward a specific category.
* The data must be structured to prevent the model just predicting final answers, and promote the model learn structured reasoning rather.

**2. Data Preprocessing**

* **Tokenization**: Mathematical expressions must be appropriately tokenized, whilst handling symbols like +, -, =.
* **Normalization**: Convert numbers into standard formats (e.g., "25" → " twenty-five ") for consistency.
* **Handling Missing or Noisy Data**: Remove ambiguous or poorly structured problems that could mislead the model.

**3. Sequence Formatting**

* Ensure consistent input-output formatting (e.g., initial question → final answer).
* Define a standardized vocabulary, including mathematical symbols and comprehension words.
* Use paddingto manage variable-length input sequences efficiently.

**4. Training Data Size**

* Large, high-quality datasets should be produceable in order promote generalisation and introduce problem variations.

**Methodology**

Preliminary research conducted suggests a **sequence-to-sequence (Seq2Seq) encoder-decoder** model would prove a good fit for our proposed use-case given their ability to process sequential data, such as text or numerical sequences, by mapping an input question sequence to an output answer sequence. This methodology will incorporate four key components:

1. **Encoder**:
   * The encoder is a recurrent neural network (RNN), long short-term memory (LSTM) that processes the input sequence token by token.
   * Converts the input into a context vector, which represents the entire sequence in a compressed form.
   * The final hidden state of the encoder is passed to the decoder.
2. **Decoder**:
   * The decoder is a recurrent neural network (RNN), long short-term memory (LSTM) that takes the context vector generated from the encoder and generates the output answer sequence step by step.
   * Predicts each answer token sequentially, using the previous token as input.
   * An **attention mechanism** will be incorporated to allow the decoder to focus on specific parts of the input sequence rather than relying solely on the context vector.
3. **Training Process**:
   * The model will be trained over a specified number of epochs using synthesised data, and **teacher forcing** will be incorporated during training to improve learning efficiency.
   * The loss function, **cross-entropy loss**, will measure difference between predicted answers and actual answer sequences (supervised learning).
   * Optimization will be performed using **Adam** to adjust model weights.
4. **Evaluation Process**:
   * Using ***Attention Visualisation*** (heatmaps) to helps understand how the model focuses on various parts of the sequential input when generating answer output;
   * Using ***Edit-distance*** and
   * Using ***BLEU score’s***

**Tooling**

1. **PyTorch**:
   * To be used to provides deep learning capabilities with modules like ***torch.nn*** for defining layers and ***torch.optim*** for optimisation.
2. **Pandas**:
   * To be used for data loading and data manipulation purpose leveraging ***DataFrames***.
3. **Optuna** (for hyperparameter tuning):
   * To be used to automate the search for optimal hyperparameters, and help to iteratively improve model performance.
4. **Attention Mechanisms**:
   * To be used to help the model focus on the most relevant parts of the question sequence while generating each answer token, to be implemented using PyTorch’s ***nn.MultiheadAttention*** to enhance sequence alignment.
5. **Scikit-Learn** (for *K-Fold*):
   * To be used for *K-Fold* cross-validation which allow the model to be trained on different splits of data, allowing a more robust evaluation of its performance across various data configurations, to be implemented using Scikit-Learn’s native **KFold** framework.
6. **MatplotLib**:
   * To be used to create a wide range of graphs and plots in support of data and model evaluation.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **AI and Machine Learning Final Project Plan** | | | | | | | | | | | | | | | | | | | | | | | | |  |  |
|  | Gateshead College |  | Project Start: | Thu, 3/27/2025 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Spencer O'Hegarty |  | Today: | Fri, 4/11/2025 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | Display Week: | 1 |  |  | Mar 24, 2025 | | | | | | | Mar 31, 2025 | | | | | | | Apr 7, 2025 | | | | | | |
|  |  |  |  |  |  |  | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | **TASK** |  | **PROGRESS** | **START** | **END** |  | M | T | W | T | F | S | S | M | T | W | T | F | S | S | M | T | W | T | F | S | S |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Phase 1 Research model and data requirements** | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Data Analysis |  | 100% | 3/27/25 | 3/27/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Model Analysis and Choice | | 100% | 3/28/25 | 3/28/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Model Prototyping |  | 100% | 3/29/25 | 3/31/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Phase 2 Model Development** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Initial data generation function | | 100% | 4/1/25 | 4/1/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Initial model development | | 100% | 4/2/25 | 4/4/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Phase 3 Evaluation and Model Iteration** | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Initial model evaluation | | 100% | 4/5/25 | 4/6/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Model Iteration |  | 100% | 4/7/25 | 4/7/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Revised model evaluation | | 100% | 4/8/25 | 4/9/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Phase 4 project Report** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Intro and objectives |  | 100% | 4/7/25 | 4/7/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Data Desc. and EDA Findings | | 100% | 4/8/25 | 4/8/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Methodology |  | 100% | 4/9/25 | 4/9/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Results, Conclusions and Refs | | 100% | 4/10/25 | 4/10/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Review - report and code | | 100% | 4/11/25 | 4/11/25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |