# Final Project: CSE 584

Archisman Ghosh
*CSE Department*
*Penn State University*
State College, PA, USA
apg6127@psu.edu

***Abstract*—This project introduces a novel approach to identifying faulty scientific questions that a large language model (LLM) fails to answer correctly. The main contribution is curating a specialized dataset of scientifically inaccurate questions, such as those involving negative mass, superluminal speeds, or physically impossible scenarios, and comparing them against correct questions from fields of Physics, Chemistry, Mathematics, Biology, and basic Geology. The research problem addresses the challenge of automatically differentiating between faulty and valid questions based on their content and structure, with the goal of improving the accuracy and reliability of AI-generated responses, especially in scientific contexts. The motivation behind this work arises from the increasing use of LLMs in research, education, and professional applications, where maintaining factual accuracy is crucial. Despite their capabilities, LLMs often produce incorrect responses when faced with faulty or poorly posed questions. By implementing a neural network classifier that uses TF-IDF vectorization to distinguish between faulty and correct question-answer pairs, this project demonstrates the potential of automated detection in enhancing the overall performance of LLMs (specifically GPT4 in this context). The results show that the classifier can accurately classify scientific questions with an accuracy of $\sim 100\%$, providing a valuable tool for improving the reliability of AI-driven systems and fostering better human-AI collaboration in scientific fields.**

***Index Terms*—Large Language Models, Neural Networks, Multi-class Classification, Datasets**

## I. Introduction

Large Language Models (LLMs) have become integral to a wide range of applications, including question-answering (Q&A) tasks in fields like education, research, and customer support [1]. Their ability to process and generate human-like text makes them highly valuable in providing automated answers, saving time, and enhancing productivity. However, LLMs can sometimes generate inaccurate or misleading responses, especially when posed with faulty or poorly framed questions. This limitation underscores the need for methods to identify and handle erroneous questions effectively. This project addresses that need by curating a dataset of faulty scientific questions—those that involve unrealistic or physically impossible scenarios—and comparing them with scientifically accurate ones. The goal is to develop an automated system capable of differentiating between correct and faulty questions using a neural network classifier. This project aims to improve the reliability of LLMs in scientific Q&A tasks, ensuring more accurate responses in critical domains.

### A. Motivation

The growing reliance on LLMs in critical domains such as science, healthcare, and education makes it essential to ensure their reliability and accuracy. When LLMs are presented with faulty or misleading questions, they may generate scientifically incorrect responses, undermining trust in their outputs. By curating a dataset of scientifically flawed questions, this project offers a systematic way to identify and address such errors. The dataset can help train models to recognize and filter out invalid inputs, improving LLM performance and ensuring that they provide more accurate and reliable answers, especially in high-stakes environments where factual correctness is paramount.

### B. Contributions

The major contributions of this study are as follows:

1) We compose a dataset of faulty science questions that strong LLMs like GPT4 [2] fail to identify as faulty and provide solutions to.
2) We further propose the discrimination of faulty and correct science questions answered by LLMs as a classification problem and design a neural network classifier to solve it.

### C. Report Structure

Section II provides a background and related works. Section III covers the dataset. Section IV describes the procedure and architecture of the neural network classifier. Section V presents the results. We discuss the results in Section VI and conclude in Section VII.

## II. Background

### A. Large Language Models

Large Language Models (LLMs), such as GPT-4, have revolutionized natural language processing by enabling machines to understand and generate human-like text. These models are built on deep neural networks, particularly transformer architectures, which leverage attention mechanisms to process sequential data efficiently. GPT-4, a state-of-the-art LLM developed by OpenAI, utilizes a vast amount of text data from diverse sources to learn patterns, relationships, and contextual nuances in language. It has 175 billion parameters, making it one of the largest models of its kind.

In the context of Question Answering (Q&A), GPT-4 excels by generating contextually relevant and coherent answers based on the input question. The model is trained using a variant of maximum likelihood estimation (MLE), where it learns to predict the probability distribution over the next word in a sentence given the previous words. This is formalized as:

$$P(w_1, w_2, \ldots, w_N) = \prod_{i=1}^{N} P(w_i \mid w_1, w_2, \ldots, w_{i-1})$$

Where $P(w_i \mid w_1, w_2, \ldots, w_{i-1})$ is the conditional probability of a word given its context. In a Q&A task, GPT-4 analyzes the question, retrieves relevant information from its training data, and generates an appropriate response based on the context. However, GPT-4's performance can degrade when questions contain errors or unrealistic assumptions, such as implausible physical scenarios or logical inconsistencies. Since the model predicts text based on patterns in its training data rather than true understanding, it may generate answers that are internally consistent but factually incorrect. This is particularly problematic in fields like science or mathematics. Therefore, it's crucial to develop methods for detecting and filtering such faulty inputs to ensure more accurate and reliable outputs.

### B. Neural Networks

Neural networks are computational models inspired by the human brain, consisting of interconnected layers of neurons. Each layer processes input data and passes it through an activation function to learn complex patterns in the data. Neural networks are particularly powerful for tasks like classification, where they can learn to identify patterns in features automatically. In natural language processing, neural networks are used to understand and classify text based on patterns learned from large datasets, enabling tasks like question answering and sentiment analysis.

### C. TF-IDF Vectorizer

The Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer is a statistical method used to convert text into numerical representations. It calculates the importance of each word in a document relative to the entire corpus, with higher values assigned to terms that appear frequently in a document but infrequently across the corpus. This allows the TF-IDF method to highlight unique and significant words, making it particularly useful for text classification tasks. In this project, the TF-IDF vectorizer is used to transform both faulty and correct questions into feature vectors that can be processed by neural networks.

### D. Related Work

Several recent works have explored different aspects of large language models (LLMs) and their applications in text classification, especially in the field of correct evaluation of questions. A study has been conducted on multiple LLMs [3], and the FAULTYMATH benchmark dataset has been proposed to evaluate LLMs on their ability to detect logical inconsistencies in math problems. The study assesses LLMs' performance as either BLIND SOLVERS or LOGICAL THINKERS, highlighting their limitations in reasoning beyond basic mathematical operations. This paper introduces a comprehensive Linguistic Benchmark to evaluate the limitations of LLMs in tasks requiring logical reasoning, spatial intelligence, and linguistic understanding. It highlights significant gaps in LLMs' capabilities compared to human reasoning, discusses the potential of prompt engineering to mitigate errors, and stresses the need for better training methodologies and human-in-the-loop systems for more reliable enterprise applications. [4] However, a study on a dataset of faulty questions from multiple scientific domains as ours has not been evaluated yet.

## III. Dataset

### A. Faulty Questions

The dataset consists of 336 faulty questions from five major scientific disciplines: physics, chemistry, biology, mathematics, and geology. These questions are intentionally designed with erroneous assumptions or unrealistic data. Common faults include negative mass in physics, superluminal speeds in astronomy, negative volume or pH in chemistry, and inconsistent facts in biology. These faults challenge the accuracy of Language Models (LLMs) and make it difficult for them to provide correct answers. The goal is to use this dataset to train and evaluate models that can identify and correct these errors, ensuring that only scientifically valid questions are processed.

### B. SQuAD Dataset

The Stanford Question Answering Dataset (SQuAD) is a large-scale collection of well-formed, factual questions based on a set of Wikipedia articles. It contains over 100,000 question-answer pairs, designed to evaluate a model's ability to extract accurate information from text. In the second part of this project, we combine this dataset with the faulty question dataset to create a balanced set of correct and incorrect questions. This enables us to train our classifier that distinguishes between valid and faulty questions.

## IV. Procedure

### A. GPT4

GPT-4 is a large-scale language model developed by OpenAI, based on the transformer architecture. With 175 billion parameters, it is one of the most powerful models. It uses self-attention mechanisms to capture long-range dependencies, excelling in tasks like question answering, text generation, and summarization. Its large-scale and deep architecture enables contextually appropriate, coherent responses across diverse natural language tasks.

### B. Classifier Architecture

The neural network model consists of a sequential architecture with four layers (Table I). The first layer is a dense layer with 512 neurons and uses the ReLU activation function. A dropout layer with a rate of 0.5 follows to prevent

overfitting. The next two hidden layers have 256 and 128 neurons, respectively, and both use ReLU activations. Finally, the output layer has a single neuron with a sigmoid activation function, making the model suitable for binary classification tasks, outputting a probability between 0 and 1 to classify the questions as either correct or faulty.

TABLE I
NEURAL NETWORK CLASSIFIER

| Layer | Number of Neurons | Activation Function |
|---|---|---|
| Input Layer | 512 | ReLU |
| Dropout Layer | - | 0.5 |
| Hidden Layer 1 | 256 | ReLU |
| Dropout Layer | - | 0.5 |
| Hidden Layer 2 | 128 | ReLU |
| Output Layer | 1 | Sigmoid |

### C. Tokenization and TF-IDF Vectorization

In this project, the process of tokenization is achieved using the `TfidfVectorizer` from `sklearn.feature_extraction.text`. Tokenization splits text into individual units (tokens), such as words or phrases, which can be understood by the model. The `TfidfVectorizer` performs tokenization along with the computation of the Term Frequency-Inverse Document Frequency (TF-IDF), which is used to represent the text data in a format suitable for machine learning. The formula for TF-IDF for a term $t$ in a document $d$ is given by:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

where:

$$\text{TF}(t, d) = \frac{\#\ t \text{ in document } d}{\text{Total terms in document } d}$$

$$\text{IDF}(t) = \log\left(\frac{N}{df(t)}\right)$$

where $N$ is the total number of documents and $df(t)$ is the number of documents containing the term $t$. The vectorizer is configured with the limiting the number of tokens to the top 5000 most informative terms. Additionally, common stop words such as "the", "and", and "is" are ignored. The output is a sparse matrix $X$, where each row corresponds to a document and each column represents the weighted importance of a term across all documents. This transformation allows the text to be used effectively as input for machine learning algorithms. The use of TF-IDF tokenization ensures that the model focuses on important terms and minimizes the impact of frequently occurring, less informative words. This approach is especially effective in tasks such as document classification, where distinguishing relevant content is crucial.

### D. Training

The model is trained using the Adam optimizer, which adapts the learning rate during training and has been proven effective for a variety of tasks. The binary cross-entropy loss function is used as the problem is a binary classification task,

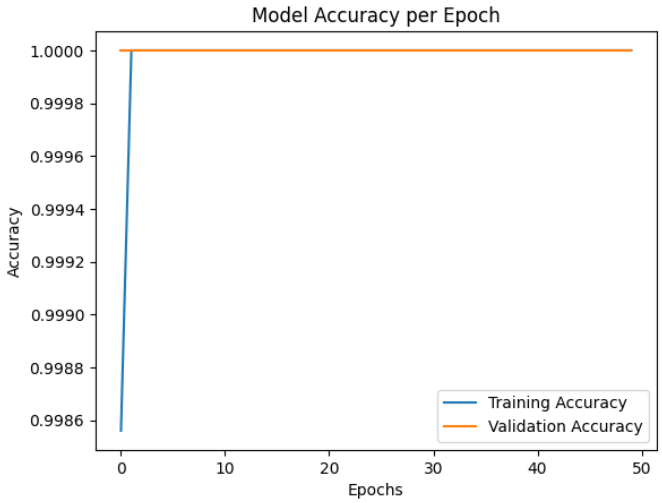


Fig. 1. Plot representing the trend in classification accuracy for the train and validation set. We can observe that the peak train accuracy is ∼100% describing the efficacy of the proposed classifier.

where the objective is to distinguish between correct and faulty questions. To prevent overfitting, an early stopping mechanism is employed, which monitors the validation loss and halts training if no improvement is observed over five consecutive epochs. This ensures that the model does not memorize the training data, and the best weights are restored once training ends. The model is trained for a maximum of 50 epochs, with a batch size of 64. A validation split of 20% is used to evaluate the model's performance on unseen data during training, allowing the model to generalize better. The training history, which includes accuracy and loss values, is tracked throughout the process to analyze the model's progress.

## V. RESULTS

### A. Simulation Setup

The classifier was implemented using TensorFlow libraries and run on a machine with 16GB RAM on an Intel Core i7-6700 CPU at a clock frequency of 3.40 GHz.

### B. Classification

We present the trends in classification in Fig. 1. From the figure we can observe that there is extremely good convergence in the training accuracy proving that the classifier works well for the dataset in classifying the faulty questions from correct ones based on the answers provided. The loss function used in the classifier is binary cross-entropy since the problem is essentially a two-class classification. It computes the negative log of the predicted probability of the positive class, meaning it penalizes the model based on how far the predicted probability for the correct class is from 1. Binary cross-entropy is calculated as

$$\mathcal{L}(y, \hat{y}) = -\left(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})\right)$$

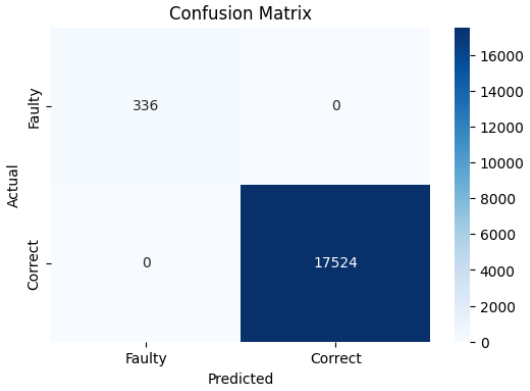


Fig. 2. Confusion Matrix for classifying faulty and correct questions. The matrix shows perfect classification, with all faulty questions correctly identified as 'Faulty' and all correct questions identified as 'Correct'. The diagonal elements represent true positives and true negatives, while there are no false positives or false negatives.

where $y$ is the true label (0 or 1), and $\hat{y}$ is the predicted probability of the positive class (output of the model, ranging between 0 and 1).

### C. Metrics

The classification report and confusion matrix provide a detailed evaluation of the model's performance along with key metrics such as precision, recall, and F1-score for each class (0 and 1), as well as overall accuracy.

*1) Precision:* For both classes, the precision is 1.00, meaning the model correctly identified all the positive and negative instances without any false positives

*2) Recall:* The recall is also 1.00 for both classes, meaning the model successfully identified all the actual instances of both the faulty and correct questions, without missing any.

*3) F1-score:* The F1-score is 1.00 for both classes, which is the harmonic mean of precision and recall, and signifies a perfect balance between these two metrics.

*4) Macro and Weighted Averages:* Both the macro and weighted averages are also 1.00, which suggests that the model performs equally well across both classes, with no class being underrepresented or misclassified.

We also observe from Fig. 2 that there are no false positives or false negatives, validating the fact that all 336 faulty and 17524 correct questions were classified correctly; leading to an overall accuracy of 100%.

## VI. Discussion

In this section, we discuss the approach to the problem, potential improvements, and future research.

### A. Dataset

The dataset could be expanded by adding more disciplines, such as computer science or social sciences, to create a more diverse testing ground. Including logical reasoning and open-ended philosophical questions would challenge GPT4 to handle ambiguity and subjective interpretation. Additionally, incorporating multi-turn or context-dependent questions could better evaluate its ability to manage complex reasoning and long-term memory. These changes would offer a more comprehensive test of GPT4's capabilities beyond simple fact retrieval. The SQuAD dataset, while valuable for fact-based QA, falls short in testing LLMs on inconsistent or logically flawed questions, which are essential for this classification task. It lacks questions involving contradictions, faulty assumptions, or philosophical nuances. To enhance the classification model, datasets like HotpotQA (multi-hop reasoning), BoolQ (yes/no questions), and PIQA (physical reasoning) can introduce diverse, complex queries. Incorporating Common Crawl and OpenWebText would add context-rich, real-world questions, while ARC (AI2 Reasoning Challenge) tests multi-step logical reasoning, providing a broader and more challenging testing environment for LLMs.

### B. Classifier

A simple neural network (NN) architecture was chosen for its ease of implementation and efficiency in solving the binary classification task. Given the relatively small dataset, a simple model with fewer layers reduces the risk of overfitting while still capturing the necessary patterns for distinguishing between faulty and correct questions. Additionally, it allows for faster training and evaluation, making it an ideal starting point for this experiment.

The NN with three dense layers, performed exceptionally well, yielding 100% accuracy on both training and test sets. This high performance, however, raises concerns about potential overfitting, data leakage, or model simplification. One possible reason for such perfect accuracy is that the model might have memorized the training data instead of generalizing from it, especially if the dataset is not diverse or large enough. The inclusion of dropout layers was intended to mitigate overfitting, but the model might still have found a way to learn too well from the available data. The absence of regularization techniques like L2 regularization or batch normalization may have contributed to this phenomenon, enabling the model to memorize rather than generalize. Another reason could be the relatively simple architecture of the model, with only three layers. While simplicity can often lead to faster training and efficiency, it may also prevent the model from fully capturing the complex relationships between input features, especially when handling diverse question types, such as faulty ones with logical inconsistencies.

### C. Future Work

To improve on this project idea, we could extend the study in a plethora of ways:

*1) Dataset:* The current dataset of 336 questions can be improved by increasing its size and diversity. Expanding to more disciplines such as computer science, economics, and social sciences would provide broader coverage. Enhancing the quality of faulty questions is essential, introducing more realistic errors like logical contradictions, ambiguous phrasing, and common misconceptions. Additionally, varying the types of errors, such as misinterpretations or inconsistent assumptions,

would increase the complexity and provide more challenging scenarios for LLMs, leading to better performance in real-world applications.

*2) Classifier:* While the simple neural network architecture provided good results, exploring more advanced models could yield even better performance. Convolutional Neural Networks (CNNs) could help capture local patterns in the text, while Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks may better understand the context and dependencies within sequential text. Pre-trained transformer models like BERT [5] or GPT4 itself, fine-tuned on this dataset, could further enhance accuracy by leveraging their extensive language understanding. These approaches might improve the model's ability to classify more complex questions.

*3) Workflow:* This study offers exciting possibilities for future research, especially in identifying the limitations of LLMs when handling faulty questions. By combining textual data with multimodal elements (e.g., images or diagrams), the model could tackle more complex questions. Enhancing model interpretability will improve trust in decision-making, allowing users to understand why a question is flagged as faulty.

## VII. Conclusion

In conclusion, this project demonstrates a novel approach to improving large language models (LLMs) by addressing their limitations in handling faulty scientific questions. Using a specialized dataset of inaccurate queries across various disciplines, we developed a neural network classifier with TF-IDF vectorization to accurately distinguish between valid and faulty questions. The classifier achieved near-perfect accuracy, highlighting the potential for automated error detection in LLMs. This work emphasizes the need for grounding LLMs in human-like reasoning, working on more reliable AI systems.

## References


[1] Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. Knowledge conflicts for llms: A survey, 2024.

[2] OpenAI et al. Gpt-4 technical report, 2024.

[3] A M Muntasir Rahman, Junyi Ye, Wei Yao, Wenpeng Yin, and Guiling Wang. From blind solvers to logical thinkers: Benchmarking llms' logical integrity on faulty mathematical problems, 2024.

[4] Sean Williams and James Huckle. Easy problems that llms get wrong, 2024.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.