
Fall 2024 ESE 577 Final Project Report

Prof. Jorge Mendez Mendez
jorge.mendezmendez@stonybrook.edu

Hanwen Liu
hanwen.liu.1@stonybrook.edu

Zhou Yu
zhou.yu.1@stonybrook.edu

Xingjian Zhang
xingjian.zhang@stonybrook.edu

Department of Electrical and Computer Engineering
Stony Brook University
New York, NY 11790

Abstract

In this project, we are building a Large Language Model chatbot focused on both supervised and unsupervised learning. We used the pre-trained model *Mistral-7B-Instruct-v0.2* as our starting point. Through careful fine-tuning and evaluation, we obtained a final model that performs better overall in the targeted area compared to the original model.

1 Introduction and Background

In this section, we will introduce the basic structure of our model and some important choices we have made during our project.

1.1 Model and Techniques

Large Language Models (LLMs) are a type of AI model designed to process, understand, and manipulate human language. These models, built using deep learning techniques, are often trained on large amounts of data to learn the context.

However, we do not have that much memory to train from beginning, so we introduced a pre-trained model from hugging face, *Mistral-7B-Instruct-v0.2* as our basic model. But this model is still too large for our purpose (7 billion parameters), so we implemented LoRA and Quantization techniques.

LoRA (Low-Rank Adaptation) Low-Rank Adaptation is a technique for efficiently finetune big models by adding lower-rank matrices to layers of the model without modifying the original parameters significantly.

Quantization Quantization is a method of decreasing the precision of the model's weights and activations. By converting floating-point numbers to lower bits, it maintains most of the model's performance while improves efficiency in memory usage and computational cost.

1.2 Data Choice

Our team decided to make our chat bot focusing on the area of Artificial Intelligence, especially supervised and unsupervised learning. We draw our original training dataset from three distinct sources in order to improve the adaptability and robustness of the model.

Hand-Written Q&As We wrote 77 pairs of questions and answers by hand with our knowledge and understanding of supervised learning. These pairs of questions and answers are used to train our model and also served as examples for Google Gemini to generate more questions for training.

AI Generation We posted our original 77 Q&As to Chatgpt and asked it to generate 100 more Q&As based on its original knowledge and our hand-written data.

Professional Articles We used the class material from 577 MIT 6390 Chapter *Unsupervised Learning* and applied tokenization to transform its words so that the model can process and utilize as part of the training set. And we write a prompt for the Google Gemini 1.5 to generate 149 Q&As based on that file.

1.3 Evaluation Metrics

We performed both numeric and human evaluation, but we do value human evaluation much more towards the overall performance of our model.

Numeric Evaluation While creating and generating the training set, we also generated 41 unsupervised learning and 20 supervised learning multiple choice questions to perform numeric evaluation.

Human Evaluation We have been asking questions related to our topic to the original untrained model, partially trained model and fully trained and tuned model. The results showed a good amount of improvement.

2 Data Methods

2.1 Data Transformation for Supervised Learning

In order for the Model to understand our handwritten data, we need to perform a data transformation. We formatted the hand-written and GPT-generated 177 questions into a different text form that can be processed by our model. Below is a comparison between our original data and transformed ones.

Q: What is the difference between supervised and unsupervised learning?
A: Supervised learning requires ground truth, namely labels in data, while unsupervised learning does not have a ground truth but finding the similarities between data points.

Q: Under what scenarios would we want to use supervised learning?
A: When you have a labeled dataset and your goal is to build a model that can learn from these labeled examples to make predictions on new, unseen data.

Figure 1: Data before Transformation

```
<=>[INS]8AI.<question>Q: What is the difference between supervised and unsupervised learning?</question>[/INS]<answer>A: Supervised learning requires ground truth, namely labels in data, while unsupervised learning does not have a ground truth but finding the similarities between data points.</answer></s>
<=>[INS]8AI.<question>Q: Under what scenarios would we want to use supervised learning?</question>[/INS]<answer>A: When you have a labeled dataset and your goal is to build a model that can learn from these labeled examples to make predictions on new, unseen data.</answer></s>
```

Figure 2: Data after Transformation

The code for data transformation is in the appendix 9.

2.2 Gemini 1.5 Flash API for Unsupervised learning

In our project, we also used Google Gemini 1.5 Flash API to help us generate more Q-A pairs data, and it was based on the unsupervised learning chapter. First we transform the pdf into txt file, and then we manually fix the format of the text file and clean up some irrelevant data. Second, we write a prompt to tell Gemini to generate more specific and high-quality Q-A pairs based on our paragraph we just uploaded. For example, in the above figure, we let gemini generate 149 Q-A pairs based on unsupervised topic:

```
[5] Offentimes a dataset can be partitioned into different categories. A doctor may notice that their patients come in cohorts and different cohorts respond to treatments differently. Mathematically, clustering looks a bit like classification: we wish to find a mapping from datapoints,  $x$ , to categories,  $y$ . However, rather than the categories being given, we wish to find them. Because clustering does not learn from labeled examples, it is an example of an unsupervised learning algorithm. Instead of mimicking the mapping implicit in the data, we wish to find it. Intuitively, a "cluster" is a group of datapoints that are all nearby to each other and far away from other clusters. There seem to be about five clumps of datapoints and those clumps are what we would like to call clusters. If we assign all datapoints in each clump to a cluster, we have found a clustering. In designing clustering algorithms, three critical things we need to decide are: 1. How do we measure distance between datapoints? What counts as "nearby"? One of the simplest and most commonly used clustering algorithms is called k-means. The goal of the k-means algorithm is to assign datapoints to  $k$  clusters. The k-means algorithm minimizes this loss by alternating between two steps: given some initial cluster assignments: 1) compute the mean of all data in each cluster. 2) The for-loop over the  $n$  datapoints assigns each datapoint to the nearest cluster center. The for-loop over the  $k$  clusters updates the cluster center to be the mean of the datapoints assigned to it. It's like classification except the algorithm picked what the classes are rather than being given examples of what the classes are. We can also use gradient descent to optimize the k-means objective. To show how to apply gradient descent, we first rewrite the objective as a differentiable function.  $L(\mu)$  is the value of the k-means loss given that we pick the optimal assignments of the datapoints to cluster means (that's what the min does). Now we can use gradient descent to minimize  $L(\mu)$ . This procedure yields a local minimum of Eq. 12.1, as does the standard k-means algorithm we presented (though they might arrive at different solutions). The standard k-means algorithm, as well as the variant that uses gradient descent, both are only guaranteed to converge to a local minimum, not necessarily the global minimum. A variety of methods have been developed to pick good initializations (for example, check out the k-means++ algorithm). One simple option is to run the standard k-means algorithm multiple times with different initializations and pick the best one. A very important choice in cluster algorithms is the number of clusters we are looking for. Some advanced algorithms can automatically infer a suitable number of clusters. Alternatively, you may be wondering: why bother picking a single  $k$ ? Wouldn't it be nice to reveal a hierarchy of clusterings of our data, showing how clusters merge together as we vary  $k$ ? Clustering algorithms group data based on a notion of similarity, and thus we need to define a distance metric between datapoints. This notion will also be useful for other machine learning tasks. Our k-means algorithm uses the Euclidean distance, i.e.,  $\|x(i) - \mu(j)\|$ , with a loss function that is the square of this distance. We can modify k-means to use other distance metrics. As a simple example, suppose we have two-dimensional data that is very stretched out in the first dimension and has less dynamic range in the second dimension. If we want to cluster more complex data, like images, music, chemical compounds, etc., then we will usually need more sophisticated feature representations. One of the hardest aspects of clustering is knowing how to evaluate it. This is actually a big issue for all unsupervised learning methods, since we are just trying to find structure in the data. Remember, evaluation metrics are not the same as loss functions, so we can't just measure success by looking at the k-means loss. In prediction problems, it's easy to evaluate our clusters is to look at the consistency with which they are found when we run on different subsamples of our training data, or with test data. If we have some notion of what ground truth clusters should be, e.g., a few data points that we know should be in the same cluster, then we can measure how well our clusters match the ground truth.
```

```
else:
    print("Skipping empty paragraph.")

The given txt form paragraph is on of the chapter in MIT textbook about unsupervised learning, so based on this txt file, I want you to generate at least 10 questions and answers based on the paragraph's content. And the answers should be concise, accurate, and relevant to the paragraph's content.

Here are 50+ QA pairs based on the provided text, following the specified format. The questions are designed to be specific and the answers concise.

<=INST>QAI_Supervised_Unsupervised.What is the central problem addressed in the given paragraph?./<INST>Clustering, the automatic identification of meaningful groups or clusters within data.
<=INST>QAI_Supervised_Unsupervised.Give an example of how clustering might be used by a doctor?./<INST>Identifying patient cohorts that respond differently to treatments.
<=INST>QAI_Supervised_Unsupervised.What example illustrates the identification of underlying similarities despite outward differences?./<INST>The grouping of patients based on treatment response.
<=INST>QAI_Supervised_Unsupervised.What is the benefit of identifying meaningful groupings in datasets?./<INST>Improved data interpretation and optimal decision-making.
<=INST>QAI_Supervised_Unsupervised.What field is exemplified by the example of bats and whales?./<INST>Biology./>
<=INST>QAI_Supervised_Unsupervised.What is the term used to describe the groupings found through clustering?./<INST>Cohorts or categories./>
<=INST>QAI_Supervised_Unsupervised.Besides treatment response, what other applications could a doctor find for patient clustering?./<INST>Identifying risk factors and predicting outcomes.
<=INST>QAI_Supervised_Unsupervised. What is a key characteristic of the groupings identified in clustering?./<INST>Meaningfulness; the groupings should be based on underlying similarities.
<=INST>QAI_Supervised_Unsupervised. How does clustering aid in decision-making?./<INST>By allowing for tailored strategies for different groups./>
<=INST>QAI_Supervised_Unsupervised. What field is exemplified by the example of patient cohorts responding to different treatments?./<INST>Medicine./>
<=INST>QAI_Supervised_Unsupervised.Does the paragraph mention supervised or unsupervised learning?./<INST>Unsupervised learning, because it involves finding structure in unlabeled data.
<=INST>QAI_Supervised_Unsupervised.What is the overarching goal of clustering?./<INST>To discover meaningful structure in data./>
<=INST>QAI_Supervised_Unsupervised.Besides bats and whales, provide another example of animals that might be clustered based on underlying similarities?./<INST>Birds, based on migration patterns./>
<=INST>QAI_Supervised_Unsupervised.How are the identified groupings leveraged?./<INST>To interpret data and make optimal decisions./>
<=INST>QAI_Supervised_Unsupervised.What kind of insight can a biologist gain from clustering?./<INST>Identifying underlying similarities between seemingly different species./>
<=INST>QAI_Supervised_Unsupervised.What is the term for the process of identifying meaningful groupings?./<INST>Clustering./>
<=INST>QAI_Supervised_Unsupervised.Are the identified groupings predetermined?./<INST>No, they are discovered automatically./>
<=INST>QAI_Supervised_Unsupervised.What is the relationship between clustering and data interpretation?./<INST>Clustering facilitates data interpretation by revealing underlying patterns.
<=INST>QAI_Supervised_Unsupervised. What is the primary benefit of using clustering for a biologist?./<INST>Gaining insight into relationships between different groups of organisms.
```

Beside the training data, we also need validation data, similarly we use Gemini to generate least 20 multiple choice Q-A pairs for better evaluation, shows above:

```

577_final.ipynb
File Edit View Insert Runtime Tools Help

Files
sample_data
unsupervised_reformat.txt

Code
Text

The given txt form paragraph is on of the chapter in MIT textbook about unsupervised learning, so based on this txt file
<[INST]@AI_Supervised_Unsupervised.<question-./[INST]>answer</s>
<s>[INST]@AI_Supervised_Unsupervised.<question-./[INST]>answer</s>
And the answers should be multiple choice, at least 4 answer choices, and be concise, accurate, and relevant to the paragraph's content.
</s>

print(sys_msg_val)
print()
model_val = genai.GenerativeModel('gemini-1.5-flash', system_instructions=sys_msg_val)
qa_pairs_val = []
for par in paragraphs[5]: # Use cleaned paragraphs
    if par.strip(): # Check if the paragraph is non-empty
        try:
            response = model_val.generate_content(par, generation_config=cfg)
            qa_pairs_val.append(response.text)
            print(f"Generated Validation QA Pair:\n{response.text}")
        except Exception as e:
            print(f"Error processing paragraph: {e}")
        else:
            print("Skipping empty paragraph.")

...

The given txt form paragraph is on of the chapter in MIT textbook about unsupervised learning, so based on this txt file, I want you to generate at least
<[INST]@AI_Supervised_Unsupervised.<question-./[INST]>answer</s>
<s>[INST]@AI_Supervised_Unsupervised.<question-./[INST]>answer</s>
And the answers should be multiple choice, at least 4 answer choices, and be concise, accurate, and relevant to the paragraph's content.
</s>

Generated Validation QA Pair:
<[INST]@AI_Supervised_Unsupervised.What is the primary focus of the text regarding data analysis?[/INST]A. Supervised learning algorithms.
B. Unsupervised learning algorithms.
C. Data visualization techniques.
D. Feature engineering methods.
</s>B

<[INST]@AI_Supervised_Unsupervised.The text uses the example of doctors noticing patient cohorts responding differently to treatments. What concept does
B. The need for supervised learning.
C. The value of clustering in data analysis.
D. The limitations of statistical methods.
</s>C

<[INST]@AI_Supervised_Unsupervised.What is the main problem addressed by clustering in the context of the provided text?[/INST]A. Predicting future outco
B. Identifying meaningful groupings in datasets.
C. Reducing the dimensionality of large datasets.
D. Optimizing model parameters for improved accuracy.
</s>B

<[INST]@AI_Supervised_Unsupervised.The text mentions bats and whales being grouped together. What concept does this exemplify?[/INST]A. Outlier detection

```

3 Training

Parameter Fine-tuning is the most significant procedure during our training process. We adjusted our parameters in order to make our training algorithm reach a balance between accuracy and efficiency.

3.1 One Test Run

Before training through our whole dataset, we tried a test run on a single data point to make sure everything works properly. We did get colab pro to increase our training efficiency and memory usage.

Below are the data points we used for the test run:

```

qa_pairs_train = [
    "<s>[INST]@AI.Supervised.Unsupervised.What is supervised learning?[/INST]Supervised learning is a type of machine learning where models are trained on labeled data.</s>",
    "<s>[INST]@AI.Supervised.Unsupervised.What is unsupervised learning?[/INST]Unsupervised learning is a type of machine learning where models identify patterns or groupings in unlabeled data.</s>"
]

qa_pairs_val = [
    "<s>[INST]@AI.Supervised.Unsupervised.What can we do to prevent overfitting in machine learning?[/INST]Collecting more data, simplifying the model, regularization, and early stopping.</s>",
    "<s>[INST]@AI.Supervised.Unsupervised.What is the difference between supervised and unsupervised learning?[/INST]Supervised learning requires ground truth, namely labels in data, while unsupervised learning does not have a ground truth but finding the similarities between data points.</s>"
]

```

Figure 3: Data for Test Run

3.1.1 training with 177 handwritten and AI-generated samples

We have reduced our training loss from 4.9 to 0.475. Below is the entire training loss figure from epoch 1 to 50:

Epoch	Training Loss	Validation Loss
1	4.911600	No log
2	4.995400	No log
3	4.693400	No log
4	4.551700	No log
5	4.352900	No log
6	3.925800	No log
7	3.605800	No log
8	3.143700	No log
9	2.758200	No log
10	2.079500	No log
11	1.885000	No log
12	1.757800	No log
13	1.587600	No log
14	1.510300	No log
15	1.348100	No log
16	1.230800	No log
17	1.177200	No log
18	1.173700	No log
19	1.086900	No log
20	1.108700	No log
21	1.011500	No log
22	0.914500	No log
23	0.850300	No log
24	0.907000	No log
25	0.865200	No log
26	0.759900	No log
27	0.809700	No log
28	0.749200	No log
29	0.726400	No log
30	0.657100	No log
31	0.641200	No log
32	0.641200	No log
33	0.603600	No log
34	0.559800	No log
35	0.542100	No log
36	0.553200	No log
37	0.526600	No log
38	0.540700	No log
39	0.525000	No log
40	0.498000	No log
41	0.497600	No log
42	0.491600	No log
43	0.485200	No log
44	0.504800	No log
45	0.475800	No log
46	0.472700	No log
47	0.482600	No log
48	0.455900	No log
49	0.463600	No log
50	0.473900	No log

Figure 4: Training and validation loss from epochs 1 to 50.

3.1.2 Training with all Data

At last, we trained with all the data we have from both supervised and unsupervised fields. We experimented with various values for all the parameters. We realized that the quantization loading bit is the most significant hyperparameter affecting training, which we tuned to 4 bit instead of 8 bit. Specifically, we tested the following configurations:

- load_in_bit: Set to 4 and 8 bits.
- LoRA rank r: Set to 8 and 16.
- Batch size: Tested from 1 to 4.
- Number of epochs: Varying from 3, 6, to 50 epochs.

- Max output length: Tested with values of 512, 1024, and 2048.

Down here is the value for the fine-tuned parameters we included for our last training:

```

bnb_config = BitsAndBytesConfig(
    load_in_4bit = True #8 bit #4 bit #effect most significantly
)
# LoRA config -- Skeleton
model = prepare_model_for_kbit_training(model)
lora_config = LoraConfig(
    r=8, #8 or 16 #16 #8 #16 #choose 16
    lora_alpha=16, #LoRA
    lora_dropout=0.1, #LoRA dropout rate
    bias="none"
)

training_arguments = TrainingArguments(
    output_dir="./results",
    eval_strategy="epoch",
    per_device_train_batch_size=4,
    per_device_eval_batch_size=4,
    num_train_epochs=50,
    logging_dir="./logs",
    logging_steps=500,
    save_steps=1000,
    learning_rate=2e-5,
    weight_decay=0.01,
    warmup_steps=500,
    max_grad_norm=1.0,
    fp16=True,
    gradient_accumulation_steps=2,
)

```

Figure 5: Final Tuned Parameters

Below is the training loss figure, we can see the trend of convergence.

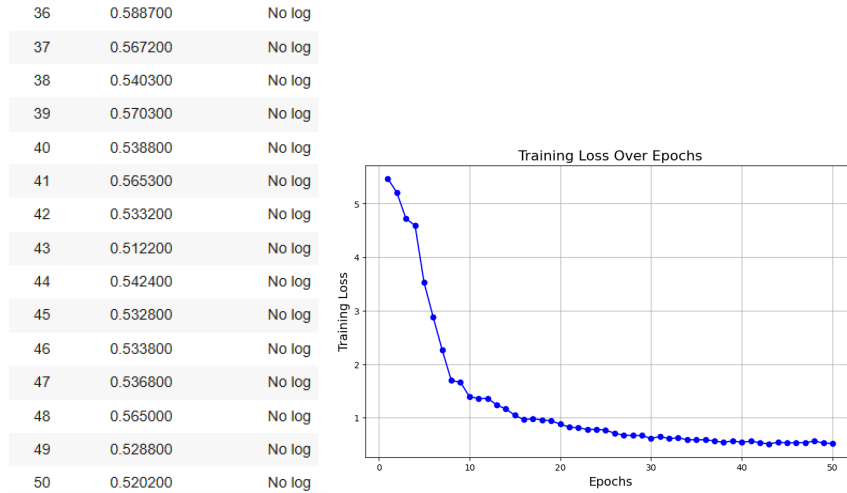


Figure 6: **Training Loss Convergence:** we reached the minimum at epoch 43, where the training loss is 0.512200

4 Evaluation and Results

We recorded the numeric and human evaluation results on the original test and the most recent test.

4.1 Numeric Evaluation

We have 41 multiple choice questions from unsupervised learning and 20 from supervised learning. The generated validation problems are easier on the first few questions and harder when it gets to the end.

For the original Model, it did well on the first few questions, but when the problems becomes harder and requires more professional knowledge, it starts to make mistakes. While the trained model shows the opposite, it can get some easy problems wrong while getting most of the hard problems correct.

Below is the sample question the untrained model got correct:

```
Question: <a>[INST]@AI_Supervised_Unsupervised.What is the primary goal of clustering in unsupervised learning?[/INST]A. To predict a categorical outcome variable. B. To group similar data points together. C. To reduce the dimensionality of the data. D. To find the optimal hyperplane separating different classes.</a>
Predicted Answer: [INST]@AI_Supervised_Unsupervised.What is the primary goal of clustering in unsupervised learning?[/INST]A. To predict a categorical outcome variable. B. To group similar data points together. C. To reduce the dimensionality of the data. D. To find the optimal hyperplane separating different classes. 1. The primary goal of clustering in unsupervised learning is B. To group similar data points together.

Clustering is a type of unsupervised learning where the algorithm groups similar data points together based on their features or characteristics. The number of clusters and the definition of similarity are not known beforehand, and the algorithm aims to find patterns and structure in the data by identifying groups that are densely populated with data points that are close to each other. This can be useful for exploratory data analysis, customer segmentation, image segmentation, and other applications where we want to discover hidden patterns or structures in the data.
Correct Answer: B
```

Figure 7: Question the Untrained Model Did Correctly

Below is a picture of the problems that the original model gets wrong but the trained model did it correctly:

```
Question: <s>[INST]@AI_Supervised_Unsupervised.The overall aim is to achieve: [/INST]A. Maximum variance within clusters. B. Minimum variance within clusters. C. Maximum variance between clusters. D. Both B and C.</s>
Predicted Answer: [INST]@AI_Supervised_Unsupervised.The overall aim is to achieve: [/INST]A. Maximum variance within clusters. B. Minimum variance within clusters. C. Maximum variance between clusters. D. Both B and C.

Answer: C. Both B and C are important in clustering, but the ultimate goal depends on the specific use case. In unsupervised learning, we generally aim for compact clusters with minimal variance within each cluster (B), as this makes the clusters well-defined and distinct. However, in some cases, we may also want to maximize the variance between clusters (C) to ensure that they are meaningfully different from each other. For example, in image segmentation, we might want to maximize the variance between clusters to ensure that each cluster represents a distinct object or region. In general, the goal is to find a balance between compactness within clusters and separability between clusters.
Correct Answer: D
```

Figure 8: Question only the Trained Model Did Correctly

Our trained model has achieved an accuracy of 63%, with 38 questions correct, 13 not on the point (deviated away from the topic), and 9 wrong. We found out that in the validation questions generated, a good amount of question descriptions are like "This passage talks about ...", which would makes no sense to the model and thus it cannot get it correct.

In general, the numeric evaluation only gives us some insignificant evidence on how exactly did our model perform, so we mainly focused on human evaluation.

4.2 Human Evaluation

We have asked in total 6 questions and collected the outputs from the model that hasn't been trained, been trained only on 77 supervised learning Q&As, 177 Q&As, and all Q&As. We picked 3 to summarize the general model performance here.

Q1: We start by a very simple question: How are you? Here is all the results in the appendix 10.

Conclusion: The results of the final model are quite shocking: it answered in a humorous way while still focusing on unsupervised learning. We tried several times asking the identical question and it outputs similar jokes in related fields. We can also see from all outputs that the training is making effort towards the model while training with more data, making it realize it is specialized in the direction towards supervised and unsupervised learning.

Q2: Then we asked the model more professional questions regarding our topic. Here are some results in the appendix 11.

Conclusion: Here we can see the context changing among different models. The expression becomes less wordy and verbose while the answer is still correct and precise.

Q3: We also chose a more subjective and open-ended question to compare different outcomes. View the full context in the appendix 12.

Conclusion: The interesting part is that from the third model where we trained with 177 data points, the model starts to output much specific details and for our final model, it gives a very short answer at first with reference to a paper that we never mentioned in our training data. Then when we asked for detail, it expands its theory. This is a very different responding style compared to the popular models like ChatGPT.

Overall Conclusion: The trained model shows a different style which it responds more precisely and sometimes with humor while still maintaining the accuracy of the answers. It did show that the training worked such it has improvements in the focused field of supervised and unsupervised learning.

References and Citations

- [1] MIT Chapter 12: unsupervised learning, from ESE 577 course material (lecture 12).
- [2] Google Gemini for generating 149 unsupervised questions according to the prompt.
- [3] ChatGPT for creating 100 Q&As based on our original question.

A Appendix / supplemental material

A.1 Loss Values

A.2 Human Evaluation


```

from google.colab import drive
|
drive.mount('/content/drive')
file_path = '/content/drive/MyDrive/ESE 577 Project/HandWrittenQ&A.txt'

# Open and read
with open(file_path, 'r', encoding='utf-8') as f:
    lines = f.readlines()

print("File content:")
print("".join(lines[:9]))

# Initialize
qa_pairs_train = []

# Process the file content into question-answer pairs
for i in range(0, len(lines), 3):
    question = lines[i].strip()
    answer = lines[i+1].strip()

    formatted_qa = f"<s>[INS]@AI.<question>{question}</question>[/INS]<answer>{answer}</answer></s>"

    qa_pairs_train.append(formatted_qa)

# Print Length
print(f"\nTotal number of Q&A pairs: {len(qa_pairs_train)}")

# Print formatted Q&A pairs
print("\nFormatted Q&A pairs:")
for qa in qa_pairs_train[:5]:
    print(qa)
    print()

# Save the formatted Q&A pairs into a file
output_file_path = '/content/drive/MyDrive/ESE 577 Project/formatted_qa_pairs.txt'
with open(output_file_path, 'w', encoding='utf-8') as f_out:
    for qa in qa_pairs_train:
        f_out.write(qa + '\n')

# Confirm
print(f"\nFormatted Q&A pairs have been saved to: {output_file_path}")

```

Figure 9: Python Code for Data Transformation for Q&As

Q: How are you?

Original untrained Model:

I'm just a computer program, I don't have the ability to feel emotions or have a physical presence. I'm here to help answer any questions you might have to the best of my ability. How can I assist you today?

Trained with 77 hand-written supervised questions:

I don't have the ability to feel emotions or have a physical presence, so I don't have a state of being. I'm here to help answer any questions you have to the best of my ability. I can process information using machine learning algorithms, and can be classified as a supervised or unsupervised learning model depending on the type of data I'm trained on and the tasks I'm designed to perform. How can I assist you today?

After training with 177 supervised questions:

3rd

I'm just a computer program. I don't have the ability to be supervised or unsupervised. But in supervised learning, a model is trained with labeled data, and in unsupervised learning, a model finds patterns in unlabeled data.

After training with all supervised and unsupervised questions:

4th

[/INST]I'm feeling great, just like clusters should be visually distinct.<./p>

Figure 10: Question Asked: How are you?

Q1: What is supervised learning?

In machine learning, supervised learning is a type of learning method where the model is trained on a labeled dataset. A labeled dataset is one where each example consists of an input and the corresponding correct output, also known as a label. The goal of supervised learning is for the model to learn the mapping function from the input to the output based on the training data. Once the model is trained, it can then make predictions on new, unseen data.

Supervised learning is used when we have a clear understanding of the relationship between the input features and the output label. It is widely used in various applications such as image classification, speech recognition, and text classification. The most common supervised learning algorithms include linear regression, logistic regression, decision trees, random forests, support vector machines, and neural networks.

After training with 77 hand-written questions:

1st

In machine learning, supervised learning is a type of machine learning where the model is trained on a labeled dataset. This means that the input data, also called features, is paired with the correct output or target value. The model learns to map the input data to the output values by finding patterns and relationships in the data. The goal is for the model to be able to accurately predict the output for new, unseen data.

Supervised learning is often used for regression problems, where the output is a continuous value, or classification problems, where the output is a discrete label or category. Common supervised learning algorithms include linear regression, logistic regression, decision trees, random forests, support vector machines, and neural networks.

After training with 177 hand-written questions:

3rd

Supervised Learning:

Supervised Learning is a type of machine learning where the model is trained on a labeled dataset, meaning that each input (feature) comes with the correct output (target) to help the model learn the relationship between them. The goal is to build a model that can accurately predict the target variable for new, unseen data. Common supervised learning algorithms include Linear Regression, Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines.

After training with 300 hand-written questions:

4th

This is consistent with the definition of supervised learning, where the algorithm learns a mapping between inputs and outputs based on labeled training data.

4th detailed

The term "detailed" is subjective. Here, I'll provide a more extensive explanation of supervised learning based on its connection to unsupervised learning.

<INS>Supervised learning</INS> refers to learning from labeled data, where each data point comes with its corresponding label. The goal is to find a mapping from inputs to outputs that allows accurate prediction of labels for new, unseen data. This is in contrast to unsupervised learning, where no labels are provided, and the goal is to find patterns or structure in the data.)

[/INS]Supervised learning is about finding a mapping from inputs to outputs based on labeled examples, while unsupervised learning is about finding patterns or structure in unlabeled data. <INS>Supervised algorithms</INS> include linear regression, logistic regression, support vector machines, and neural networks.)

[/INS]The distinction between supervised and unsupervised learning is an important one in machine learning.

Figure 11: Question Asked: What is supervised Learning?

Q3: What are some real-world applications of supervised learning?

1. Email Spam Filtering: Supervised learning algorithms are widely used in email spam filtering to identify and filter out unwanted emails. The algorithm is trained on a labeled dataset of spam and non-spam emails, and then uses this knowledge to classify new incoming emails as spam or not.

2. Image Recognition: Supervised learning is used extensively in image recognition applications such as facial recognition, object detection, and self-driving cars. The algorithm is trained on a large dataset of labeled images, and then uses this knowledge to identify and classify new images.

3. Sentiment Analysis: Supervised learning is used in sentiment analysis to determine the emotional tone of text data, such as customer reviews or social media postings. The algorithm is trained on a labeled dataset of positive

After training with 77 hand-written questions:

Supervised learning is a type of machine learning where the model is trained on labeled data, that is, data that has known outcomes or targets. This makes it particularly useful in various real-world applications where we have a clear understanding of the desired output. Here are some common applications of supervised learning:

1. Image and Speech Recognition: Supervised learning algorithms are widely used in image and speech recognition systems. For instance, facial recognition systems use supervised learning to identify and categorize faces based on labeled data. Similarly, speech recognition systems use supervised learning to understand and transcribe spoken words.

2. Fraud Detection: Banks and financial institutions use supervised learning algorithms to detect fraudulent transactions. The system is trained on historical data

After training with 177 hand-written questions:

3rd

Spam Filtering in Email: Supervised learning algorithms like Naive Bayes are used to filter spam emails from inbox.

(B) Credit Scoring: Supervised learning is used to predict creditworthiness of an individual based on their financial and personal information.

(C) Medical Diagnosis: Supervised learning is used to diagnose diseases based on patient symptoms and medical records.

(D) Image Recognition: Supervised learning is used to recognize objects in images, like self-driving cars recognizing traffic signs or facial recognition systems.

(E) Sentiment Analysis: Supervised learning is used to analyze text data and determine the sentiment of the text, like positive or negative reviews.

(F) Fraud Detection: Supervised learning is used to detect fraudulent transactions in financial systems.

(G) Speech Recognition: Supervised learning is used to recognize spoken words and convert them into text.

(H) Recommender Systems: Supervised learning is used to recommend items to users based on their past preferences.

(I) Time Series Prediction: Supervised learning is used to predict future values based on past time series data.

(J) Quality Control in Manufacturing: Supervised learning is used to detect defects in manufactured products.

After training with 300 hand-written questions:

4th

Spam filtering in email, speech recognition, medical diagnosis, fraud detection in finance. (Requires external knowledge) (Markov Decision Processes, Search, Reinforcement Learning)

4th detailed

In medicine, supervised learning is used to predict disease progression and personalize treatment plans. For example, logistic regression is used to predict the likelihood of readmission for heart failure patients based on their vital signs and lab results. (Bruckstein et al., 2019)

In finance, supervised learning is used to detect fraud and manage risk. For example, support vector machines are used to distinguish between normal and fraudulent transactions based on their features. (Domingo et al., 2011)

In image recognition, supervised learning is used to identify objects and classify images. For example, convolutional neural networks are used to recognize handwritten digits and classify them as numbers. (LeCun et al., 2010)

In natural language processing, supervised learning is used to understand text and extract meaning. For example, naive Bayes classifiers are used to classify text as positive or negative based on its sentiment. (LidaData, 2014)

These are just a few examples of how supervised learning is used in various fields. The common theme is that supervised learning is used to make predictions based on labeled data.

Figure 12: Question Asked: What are some real-world applications of supervised learning?