Fine-Tuning BERT for Question-Answering: Documentation Report

1. Methodology and Steps Taken

1.1 Importing Required Libraries

The following key libraries were used to set up the environment:

- **evaluate** for computing evaluation metrics.
- **datasets** for loading and processing the SQuAD dataset.
- **transformers** for utilizing BERT and fine-tuning it for question-answering.
- **Trainer** and **TrainingArguments** for managing the training process.
- **AutoTokenizer** and **AutoModelForQuestionAnswering** for model selection and tokenization.

1.2 Loading the Data

The dataset used for training is the **Stanford Question Answering Dataset (SQuAD)**, identified as **rajpurkar/squad**. The pre-trained model selected is **BERT (google-bert/bert-base-uncased)**.

The dataset contains:

- **87,599 training samples**
- **10,570 validation samples**

1.3 Data Preprocessing

Preprocessing involved the following steps:

- Tokenizing questions and contexts using the BERT tokenizer.
- Aligning tokenized data with their respective answer spans.
- Storing the start and end positions of answers in the context.

1.4 Model Fine-Tuning

The pre-trained **bert-base-uncased** model was fine-tuned using the **Trainer API** from the **transformers** library. The key steps involved:

1. **Loading the model**: **AutoModelForQuestionAnswering** was used to load the BERT model.
2. **Defining Training Arguments**: Essential parameters were specified, such as learning rate, batch size, and number of epochs.

3. **Training the Model**: The dataset was split into training and validation sets, and fine-tuning was performed using backpropagation.
4. **Saving the Model**: After training, the model was saved for later inference.

2. Experimentation Details

2.1 Hyperparameters Used

The training was conducted with the following hyperparameter choices:

- **Learning Rate**: 3e-5
- **Batch Size**: 16
- **Epochs**: 2
- **Weight Decay**: 0.01

2.2 Challenges Encountered

- **Memory Consumption**: Fine-tuning large transformer models requires high GPU memory. Optimization techniques such as gradient accumulation were considered.
- **Slow Training Speed**: Fine-tuning took significant time due to the large dataset and model size. Batch size adjustments helped manage computational efficiency.

3. Evaluation Results and Insights

The model was evaluated using standard metrics for question-answering:

- **Exact Match (EM)**: Measures how many predicted answers exactly match the ground truth.
- **F1 Score**: Measures token overlap between predictions and ground truth answers.

Results on the validation set:

- **Exact Match (EM):** 80.3%
- **F1 Score:** 89.7%

These results indicate strong performance, with high accuracy in answer extraction.