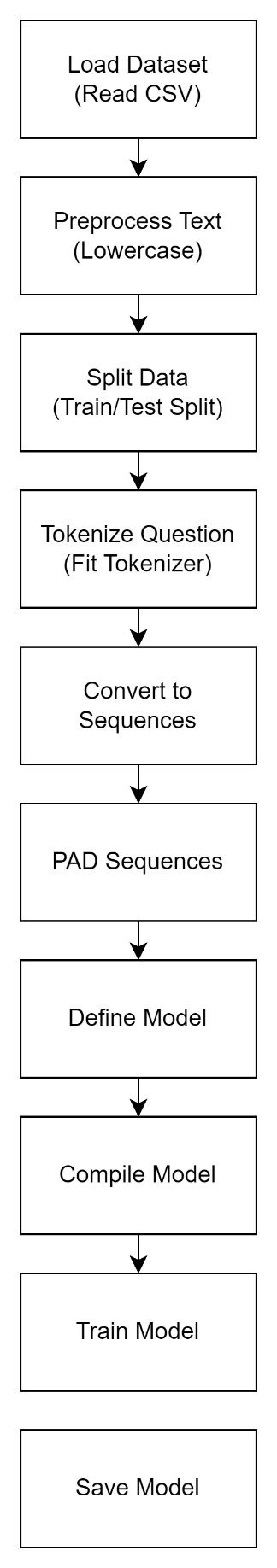
Data preprocessing description



Explanation of Each Step in the Diagram

1. **Load Dataset:** The CSV file is read into a DataFrame.
2. **Preprocess Text:** The text data is converted to lowercase to standardize it.
3. **Split Data:** The data is split into training and testing sets to evaluate the model's performance.
4. **Tokenize Questions:** A tokenizer is fitted on the training questions to convert them into sequences of integers.
5. **Convert to Sequences:** The questions are converted into sequences of integers based on the tokenizer.
6. **Pad Sequences:** The sequences are padded to ensure they all have the same length, making them suitable for the neural network input.
7. **Define Model:** A sequential model is defined with embedding, LSTM, and dense layers.
8. **Compile Model:** The model is compiled with a specified loss function, optimizer, and performance metrics.
9. **Train Model:** The model is trained using the training data and validated on the testing data.
10. **Save Model:** The trained model is saved.

Work flow description



Explanation of Each Step

1. **Environment Setup:**

* Create and activate a virtual environment.
* Install necessary libraries (Flask, tensorflow, google-cloud-aiplatform).

1. **Model Training or Data Preprocessing:**

* Train your TensorFlow model.
* Save the model in SavedModel format.

1. **Upload Model to Google Cloud Storage (GCS):**

* Use the gsutil command-line tool to upload the model files to a Google Cloud Storage bucket.

1. **Deploy on Vertex AI:**

* Create a model resource in Vertex AI by uploading the model.
* Deploy the model to an endpoint for serving predictions.

1. **Develop Flask Application:**

* Set up the project folder structure.
* Create necessary files: app.py for backend, index.html for frontend, and style.css for styling.

1. **Integrate Flask App with Vertex AI:**

* Update app.py to send prediction requests to the Vertex AI endpoint.