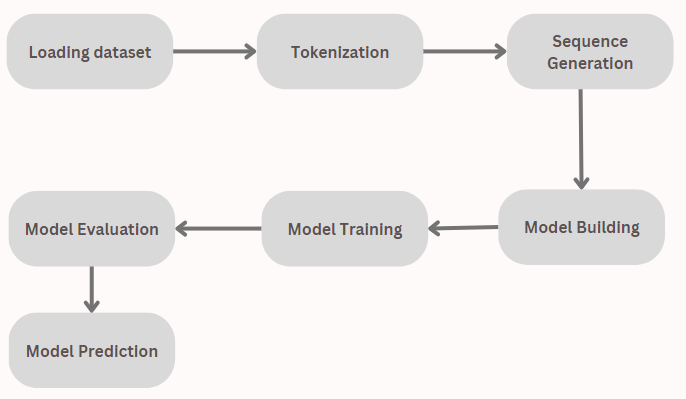
**Next Word Prediction**

Next Word Prediction is a project that aims to develop a robust language model capable of predicting the next word in a given text sequence. By utilizing deep learning techniques, specifically recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks, the project aims to learn patterns and relationships within a large corpus of text data. This prediction capability enhances text generation, improves word completion, and enables intelligent auto-suggestion functionalities in natural language processing tasks. The proposed framework of next-word prediction is:



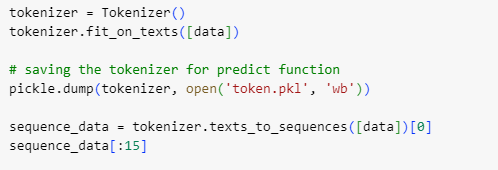
**Data Preprocessing:**

The project starts by importing the necessary libraries, including TensorFlow and Keras, which provide powerful tools for deep learning. The text data is loaded from the "Pride and Prejudice.txt" file and stored in a list, with each element representing a line from the text. These lines are then concatenated to form a single string. Preprocessing is performed to remove unnecessary characters such as newlines, carriage returns, Unicode characters, and quotation marks, which could impact the accuracy of the language model. Additionally, any extra spaces are eliminated to ensure the consistency of the data.



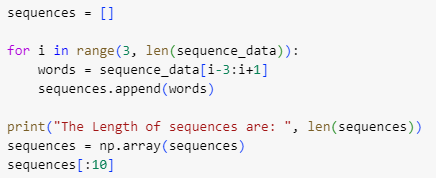
**Tokenization:**

To prepare the text data for modeling, the Tokenizer class from the Keras library is utilized. The Tokenizer is fitted on the preprocessed text data, which allows it to learn the vocabulary of the corpus. This process involves assigning a unique integer index to each word in the vocabulary, which will later be used as input to the neural network model. The fitted tokenizer is saved as a pickle file for future use during the prediction phase.



**Sequence Generation:**

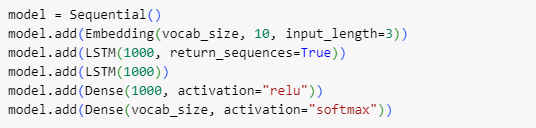
The tokenized data is further processed to generate sequences of words. This involves creating a sliding window approach where a sequence of four words is formed. The first three words in each sequence serve as input features, while the fourth word is the target output. These sequences are stored in a numpy array for efficient computation and further model training.

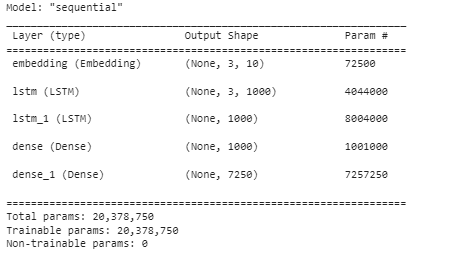


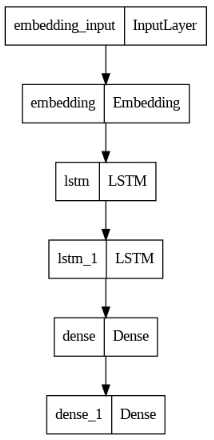
**Model Architecture:**

The neural network model for next-word prediction is constructed using the Sequential API provided by Keras. The model architecture consists of several layers. Firstly, an embedding layer is added, which maps the integer indices of words to dense vectors of fixed size. This embedding layer helps the model to learn meaningful representations of words. Next, two LSTM layers are stacked together to capture the sequential patterns in the input sequences effectively. LSTM networks are specifically designed to model the dependencies and long-term dependencies in sequential data. A dense layer with ReLU activation is then added to introduce non-linearity into the model. Finally, a dense layer with softmax activation is employed to produce a probability distribution over the vocabulary, representing the likelihood of each word being the next word in the sequence. The model's summary is printed to provide an overview of its architecture and the number of parameters.

Our model summary is:

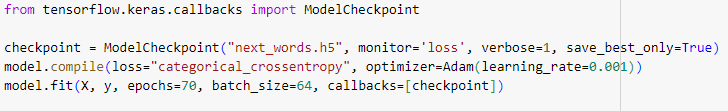






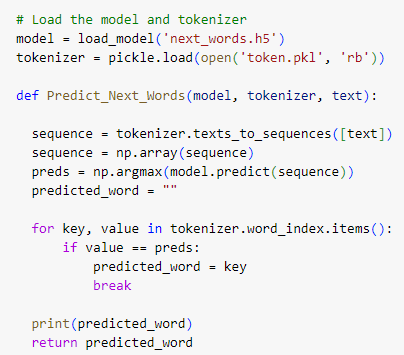
**Model Training:**

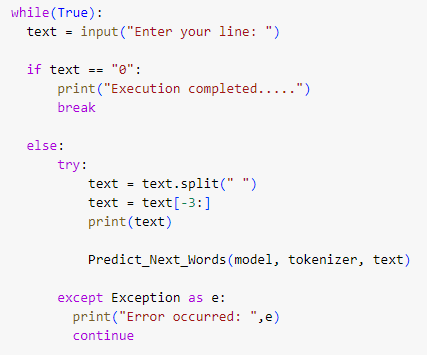
The model is compiled with the categorical cross-entropy loss function, which is suitable for multi-class classification tasks. The Adam optimizer, known for its efficiency and effectiveness in training deep neural networks, is used to update the model's weights during the training process. To monitor the model's performance, a ModelCheckpoint callback is employed to save the best-performing model based on the loss metric. The training process is initiated by fitting the model to the training data, specifying the number of epochs and batch size.



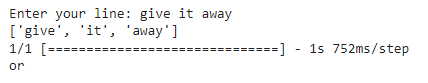
**Model Evaluation:**

Once the model is trained, it can be evaluated to assess its performance and generalization capabilities. This involves using a separate test dataset to measure metrics such as accuracy, precision, recall, and F1-score. These metrics provide insights into the model's ability to predict the next word accurately and its overall performance in understanding the language patterns within the text data.





Our prediction is:



**Next Word Prediction:**

After training and evaluation, the model is saved for future use and can be loaded along with the tokenizer. The user is then prompted to enter a line of text. The last three words from the input line are extracted, and the Predict\_Next\_Words function is called. This function takes the loaded model, tokenizer, and the extracted context as input. The function tokenizes the context, converts it into a sequence, and feeds it to the model for prediction. The model predicts the index of the next word in the vocabulary, and the tokenizer is used to retrieve the corresponding word. The predicted word is then printed as the output. The user can continue entering new lines of text, and the model will provide predictions for the next word based on the provided context.

**Conclusion:**

The Next Word Prediction project demonstrates the application of deep learning techniques, specifically LSTM-based models, for accurately predicting the next word in a given text sequence. By leveraging the power of recurrent neural networks and LSTM cells, the model can capture the contextual information and language patterns necessary for generating accurate predictions. The project offers numerous applications, including text completion, auto-suggestion, and intelligent writing assistance, which can greatly enhance user experiences in natural language processing tasks. With further advancements and fine-tuning, next word prediction models have the potential to contribute to the development of smarter and more intuitive language processing systems.