CSD TEAM - 14

NEXT WORD PREDICTION USING LSTM

TEAM DETAILS

TORATI J V CHANDRA SATISH(TL) (216Q1A4489)

KARRI JOHN (216Q1A4471)

M MANMOHAN REDDY (216Q1A4478)

M SATYA VENKATARAO (216Q1A4480)

A SURYA VARAPRASAD (216Q1A4488)

**Abstract**

Next word prediction is a critical task in natural language processing (NLP) with applications in text generation, autocompletion, and virtual assistants. This paper proposes a solution for next word prediction using Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN) designed to capture long-range dependencies in sequential data. LSTMs are particularly effective in processing and predicting words in a sequence due to their ability to remember context over time, which is crucial in language modeling.

We train the LSTM on a large corpus of text data to learn the probabilistic relationships between words. During the training phase, the model learns the structure and patterns of word sequences, enabling it to predict the most likely next word in a given context. The model is evaluated on a test set to assess its performance in terms of accuracy and relevance of the predicted words. Experimental results demonstrate that the LSTM-based approach significantly outperforms traditional methods, providing more contextually accurate and coherent predictions.

Our approach also explores the challenges of overfitting and vanishing gradients, which are common in training deep RNNs, and presents techniques such as gradient clipping and dropout to mitigate these issues. The proposed method offers promising advancements in the field of NLP, contributing to more intelligent and responsive language models in real-world applications.

**Input**

To perform next-word prediction using LSTMs (Long Short-Term Memory networks), you need a dataset of text data, preprocess it into sequences, train an LSTM model on these sequences, and then generate predictions.

**Steps for Next-Word Prediction using LSTM:**

1.**Data Preprocessing:**

Tokenize the text

Convert words to numerical sequences

Create input-output pairs

2**.Build the LSTM Model**

Define an LSTM architecture with an embedding layer

Add LSTM layers

Use a dense layer with softmax activation for prediction

3.**Train the Model**

Compile with categorical cross-entropy

Train using sequences of words

4.**Make Predictions**

Take a sequence, predict the next word, and generate text

**Process**

* + **Preprocess Text:**
    - Tokenize words
    - Convert to sequences
    - Apply padding
  + **Build LSTM Model**
    - Use an embedding layer
    - Add LSTM layers
    - Apply a dense softmax output layer
  + **Train Model**
    - Use loss and optimizer functions
    - Train on dataset
  + **Predict Next Word**
    - Input a sequence
    - Predict the next word
    - Convert output to a readable format

**Out put**

Next word prediction using an LSTM (Long Short-Term Memory) model typically involves training the model on a dataset to predict the next word in a sequence given a previous set of words.

To understand how LSTM works for next-word prediction, let me explain a general process:

* **Data Preprocessing**: Text data is tokenized into sequences of words or characters. Each word or character is then converted into numerical representations, often using techniques like one-hot encoding or word embeddings (e.g., Word2Vec or GloVe).
* **Model Training**: An LSTM model is trained to predict the next word in a sequence based on the previous words. The LSTM's architecture is designed to learn long-range dependencies in the data, which is helpful for understanding context over several words in a sentence.
* **Prediction**: After training, given an input sequence of words, the model predicts the most likely next word. The prediction is based on the learned patterns and dependencies in the training data.