Resume Categorization

Project Overview

This project aims to classify resumes into different categories based on their content using a deep learning model. The process involves preprocessing text data, training a Bidirectional LSTM model, and evaluating its performance. The project includes a script to categorize a batch of resumes and organize them into respective folders.

1. Model Selection and Rationale

Model Chosen: Bidirectional LSTM

- **Bidirectional LSTM:**
  - **Rationale:** LSTM networks are effective for sequential data because they capture long-range dependencies. A Bidirectional LSTM processes sequences in both forward and backward directions, improving context understanding and sequence learning.
  - **Embedding Layer**:
    Function: Converts word indices into dense vectors, providing a rich representation of the text.
  - **Dropout Layer**:
    Function: Prevents overfitting by randomly dropping units during training, which helps the model generalize better.
  - **Dense Layer**:
    Function: Outputs the probability distribution across categories for classification.

2. Data Preprocessing and Feature Extraction

Preprocessing Steps:

a. **Lowercasing:** Converts all text to lowercase to ensure uniformity.
b. **Remove Punctuation and Special Characters:** Eliminates non-alphanumeric characters.
c. **Remove Numbers:** Strips numbers from the text.
d. **Tokenization:** Splits the text into words.
e. **Remove Stopwords:** Removes common words that may not add value.
f. **Lemmatization:** Converts words to their base form.

Feature Extraction:

g. **Tokenization:** Converts text into integer sequences.
h. **Padding:** Standardizes sequence length for input to the model.
i. **Embedding Layer:** Transforms word indices into dense vectors.

3. Instructions for Running the Script

Install Required Libraries:

pip install tensorflow keras pandas matplotlib seaborn scikit-learn nltk joblib

Execute the Script:

python script.py <path to the resume directory>