Low Level Design Document

PROJECT TITLE:

News Category Classification

I

Introduction

:

The News Category Classification project aims to classify news articles into different categories such as sports, technology, business, entertainment, politics, etc. This project utilizes BERT (Bidirectional Encoder Representations from Transformers) model, which is a state-of-the-art language model for Natural Language Processing (NLP) tasks.

Technical Details:

The News Category Classification project will be developed using Python programming language and will utilize the following libraries:

1. Pandas: To handle and manipulate data in the form of data frames.
2. TensorFlow: To build and train the BERT model.
3. Keras: To build and compile the BERT model.
4. Hugging Face Transformers: To access the pre-trained BERT model and tokenizer.
5. Flask: To develop a web application for users to interact with the classification model.
6. HTML/CSS/JavaScript: To design and develop the user interface of the web application.

Data Preprocessing:

The first step in the project is to preprocess the data. The following steps will be performed for data preprocessing:

1. Import the dataset using Pandas library.
2. Remove any special characters, numbers, and stop words from the headlines.
3. Convert all the text to lowercase to ensure consistency.
4. Split the dataset into training, validation, and test sets.

Tokenization:

The next step is to tokenize the text using the BERT tokenizer. The following steps will be performed for tokenization:

1. Import the pre-trained BERT tokenizer from the Hugging Face Transformers library.
2. Tokenize the text using the tokenizer.
3. Convert the tokens into numerical vectors.

Model Building:

The next step is to build the BERT model. The following steps will be performed for model building:

1. Import the pre-trained BERT base uncased model from the Hugging Face Transformers library.
2. Add a dense layer with softmax activation to output the probabilities of each category.
3. Compile the model using categorical cross-entropy loss as the loss function and Adam optimizer for training.

Model Training:

The next step is to train the BERT model. The following steps will be performed for model training:

1. Fit the model on the training set using the TensorFlow and Keras libraries.
2. Validate the model on the validation set.
3. Save the model weights for future use.

Model Evaluation:

The final step is to evaluate the performance of the model on the test set. The following steps will be performed for model evaluation:

1. Load the saved model weights.
2. Predict the categories for the news articles in the test set.
3. Evaluate the performance of the model using accuracy and F1-score metrics.

Web Application Development:

The web application will be developed using Flask and HTML/CSS/JavaScript. The following steps will be performed for web application development:

1. Design the user interface using HTML/CSS/JavaScript.
2. Develop the Flask application to handle user requests.
3. Load the saved model weights in the Flask application.
4. Predict the category for the entered news article.
5. Display the predicted category on the web page.
6. Deploy the app in Azure

Conclusion:

The News Category Classification project demonstrated the effectiveness of using BERT model for NLP tasks such as text classification. The project can be further improved by using a larger dataset and fine-tuning the BERT model. This project can be used to automate the process of news categorization and can be integrated into news websites and applications.