Classifying Architectural Posts in Stack Overflow Using BERT Deep Learning Model

Group Members:
Mitul Nasit,
Arpita Poojari,
Abhijeet Desai,
Gautam Parmar,
Kiran Verma
Group: 06
Git Repo

1 Introduction

In the domain of software engineering, the process of identifying and classifying architectural posts on platforms like Stack Overflow is important to enhancing the accessibility of design-related knowledge for developers. However, the existing methods for categorizing architectural posts often lack detailed information, making it difficult to create specific tools to help the discovery of specific design insights.

To address this gap, our study focuses on the exploration of machine learning techniques within the context of architectural post-classification. Using a carefully selected dataset of categorized posts, our study focuses on applying the BERT machine learning technique, to understand and categorize architectural posts based on their purpose type. Through a systematic approach including data preparation, model development, and evaluation phases, our aim is to enhance the precision and efficiency of architectural post-classification, thereby enriching the landscape of design-centric discussions on Stack Overflow.

2 Study Design

For the BERT deep learning model assignment, the research questions are the following:

RQ1: How accurate is your model to classify architectural posts?

RQ2: How many architectural posts exist in Stack Overflow? And what are the most common types of these posts?

RQ3: What are the characteristics of the architectural posts in Stack Overflow?

Motivation: The reason for this assignment is to explore software engineering design discussions on Stack Overflow. By using advanced machine learning models, we aim to correctly categorize architectural posts and discover insights into the features of these posts. Through this analysis, we hope to improve our understanding of architectural topics on the platform and possibly identify common types of architectural posts. This assignment gives us a chance to apply the BERT deep learning technique practically and provide valuable findings that could help software engineering practices in the future.

2.1 Download the data of all posts in the dataset

To determine all the datasets we have prepared an SQL query to fetch the data of all posts from the query interface Stackoverflow. By following these steps we have Successfully fetched **2387 posts data out of 2397 provided IDs** of Programming(1628) and architectural(769) posts.

2.2 NLP pre-processing and number of terms per post

We wrote a Python script that Concatenates each post-data Question title, Question Body, and Answer Body, and then it processes and cleans the concatenated data. It utilizes the NLTK library to perform text cleaning operations including, removing punctuation and stop words as well as converting text to lowercase. The script then performs lemmatization on the cleaned text after that, in the same way, the script preprocessed text to determine the number of terms per post. Then it stored the data in the Excel file named Preprocessed_data which you can find in the dataset folder of the repository.

2.3 Classified posts into training, validation, and test set

Now to prepare a dataset for training a machine learning model, we wrote a Python script that divided our data set into a training set (80%), validation set (10%), and test set (10%). It split the preprocessed data and labels of the data into three sets also, the same script divided the data into 10 k-fold cross-validations by using the StratifiedKFold package. These datasets can be found in our repository inside the datasets folder.

3 Results

Results for Week 1 Tasks:

In the initial phase of our assignment to classify architectural posts from Stack Overflow using machine learning, a series of crucial steps were completed. Below is a detailed account of the progress made during Week 1:

3.1 Data Collection and Pre-processing

The dataset was acquired by leveraging the query interface of Stack Overflow to download essential information, including post titles, questions, and answers with the highest votes. Subsequently, a comprehensive NLP pre-processing procedure was implemented. This involved merging post titles, questions, and top-voted answers into a single string, followed by text filtering processes such as removal of HTML tags, stop words, lemmatization, stemming, and handling source code to ensure data cleanliness and uniformity.

3.2 Term Count Determination

An important metric for our subsequent deep-learning model was determined by calculating the number of terms present in each post. This information will aid in defining the input size of our model.

3.3 Data Splitting and Preparation for Model Training

Source code was developed to effectively partition the dataset into training, validation, and test sets. Employing a stratified random sampling approach ensured a well-balanced distribution of data across the sets. Additionally, provisions were made for k-cross validation, enhancing the robustness of our model training process.

3.4 Data Summary and Post-processing

A comprehensive Excel file was generated as an outcome of the pre-processing steps, containing a total of 2387 posts' data. It is essential to note that 10 posts were missing from the dataset, presumably removed or deleted from the Stack Overflow database. The Excel sheet encompassed various columns such as QuestionId, QuestionTitle, QuestionBody, AnswerId, AnswerBody, AnswerScore, preprocessed_Text, Post_ID, URL, Purpose, Solution, Programming_post, Term_Count, and Post_Type.

The successful execution of these tasks has laid a solid foundation for our subsequent endeavors in developing and training deep-learning models to classify architectural posts effectively. Week 1 has been instrumental in setting the stage for the forthcoming phases of our assignment, showcasing a structured and meticulous approach toward data collection, pre-processing, and preparation for model training.

4 Hours spent in the assignment, and the contribution of each student

Student Name	Work	Hours
Mitul Nasit	Project structure Classify posts into training, validation and test set	5
Abhijeet Desai	Data cleaning Stop word removal and lemmatization of the concatenated data	4
Arpita Poojari	Determine the number of terms per post Documentation	3
Gautam Parmar	Data fetched from stack exchanges Documentation	5
Kiran Verma	Done prework on downloading the data of all posts in the dataset Explore BERT model Documentation	4

Table 4.1: Students Report