# Automatic Question Tagging System

**Abstract**

Question-and-answer platforms like Quora and StackOverflow often require users to submit tags along with their questions to facilitate categorization and improve navigation. However, users sometimes provide incorrect or irrelevant tags, making it difficult for others to find the information they need. To address this issue, we propose an automatic question tagging system that leverages machine learning to predict relevant tags for user-submitted questions. This report details the dataset used, preprocessing steps, model implementation, and evaluation metrics.

## 1 Introduction

Question-and-answer platforms like Quora and StackOverflow often require users to submit tags along with their questions to facilitate categorization and improve navigation. However, users sometimes provide incorrect or irrelevant tags, making it difficult for others to find the information they need. To address this issue, we propose an automatic question tagging system that leverages machine learning to predict relevant tags for user-submitted questions.

## 2 Dataset Description

For this project, we used the StackSample dataset, which includes a large collection of questions and answers from StackOverflow. The dataset consists of three files:

- **Questions**: Contains the question ID, title, body, and other metadata.
- **Answers**: Contains the answer ID, question ID, body, and other metadata.
- **Tags**: Contains the question ID and the associated tags.

## 3 Technologies Used

The following technologies and libraries were used in this project:

- **Python**: The primary programming language used for data processing and model training.
- **Pandas**: For data manipulation and analysis.
- **BeautifulSoup**: For cleaning HTML tags from the text data.
- **scikit-learn**: For machine learning model implementation and evaluation.
- **NumPy**: For numerical operations.
- **SciPy**: For handling sparse matrices.

# 4 Data Preprocessing

To prepare the dataset for model training, we performed several preprocessing steps:

## 4.1 Loading the Data

We loaded the questions and tags data into Pandas DataFrames. The questions and tags data were then merged on the question ID to create a combined DataFrame containing the question text and associated tags.

## 4.2 Cleaning the Data

We removed null values and stripped HTML tags from the question titles and bodies using the BeautifulSoup library. Regular expressions were employed for further text cleaning.

## 4.3 Combining Text

The title and body of each question were concatenated into a single text field to provide a comprehensive representation of the question.

## 4.4 MultiLabelBinarizer

The tags, initially in string format, were converted into a binary matrix format using the MultiLabelBinarizer from scikit-learn. This transformation is essential for multi-label classification, where each question can have multiple tags.

# 5 Model Implementation

The goal of model training is to develop a machine learning model that can accurately predict tags for a given question. The following steps outline the training process:

### 5.1 Splitting Data

The dataset was divided into training and validation sets. The training set is used to train the model, while the validation set is used to evaluate its performance. This split helps in assessing how well the model generalizes to unseen data.

### 5.2 Vectorization

The text data was converted into numerical format using the TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer. TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents. This transformation is crucial for converting text data into a format that machine learning algorithms can process.

### 5.3 Classifier

A multi-output logistic regression model was trained to predict tags for each question. Logistic regression is a simple yet effective algorithm for multi-label classification tasks. The multi-output variant allows for predicting multiple tags for each question.

## 6 Model Evaluation

The performance of the trained model was evaluated on the validation set using various metrics:

- **Accuracy**: Measures the proportion of correct predictions out of all predictions made. While accuracy is a useful metric, it might not fully capture the performance in a multi-label setting where each question can have multiple tags.
- **F1 Score**: A harmonic mean of precision and recall, providing a balanced measure of a model's performance. Precision measures the proportion of relevant tags out of all predicted tags, while recall measures the proportion of relevant tags out of all actual tags.
- **Hamming Loss**: The fraction of tags that are incorrectly predicted. It accounts for both false positives and false negatives, making it a suitable metric for multi-label classification.

The results indicated that the model performs well in predicting relevant tags, demonstrating its potential to improve the tagging process on question-and-answer platforms.

# 7 Conclusion

The automatic question tagging system effectively addresses the challenge of incorrect and irrelevant tags by leveraging machine learning techniques. By automatically predicting relevant tags, the system enhances content organization and user experience on platforms like Quora and StackOverflow.

Future work could explore more advanced models such as deep learning approaches, incorporating additional features like user profiles and answer content, and expanding the system to other languages and domains.