CS 458/535 - Natural Language Processing
Automatic Questions Tagging System

Hifza Majeed , Mina Riaz
19P-1652 , 19P-0099

1 Problem Statement

Sites like Quora and Stack Overflow, which are created specifically to have questions and answers for their users, frequently ask their users to provide five words along with the question so that it may be readily categorized. However, occasionally people give incorrect tags, making it challenging for other users to search through. They need a system that can automatically identify the right and pertinent tags for a user-submitted inquiry to fulfill this need.

2 Motivation

For several reasons, the issue of manual question tagging is intriguing. First off, manually categorizing a huge number of questions might take a lot of time and effort, especially when there are thousands of questions to be sorted. This can result in inaccurate categorization and mistakes, which would be bad for the user experience. Second, the necessity for effective and efficient organization of user-generated material is becoming more and more crucial as its volume keeps expanding across numerous platforms. This need can be met by an automatic question tagging system, which offers a quick and precise method of classifying and arranging massive amounts of user-generated questions.

3 Example

Let's use a customer care forum example to demonstrate this issue. Consider a business that offers a forum for customer help where clients can post inquiries and receive responses from either the support team or other clients. Users may find locating the information they require challenging due to the forum's expansion and thousands of inquiries. The business creates an automatic question-tagging system that can classify each question according to its topic to resolve this problem. For instance, the automatic tagging system may apply tags like "password," "account security," and "login difficulties" to a customer's inquiry on how to change their password, making it simpler for other users to locate similar information in the future.

4 Benefits

To increase the effectiveness and efficiency of question-and-answer systems, such as online forums, knowledge bases, and customer service portals, an automatic questions tagging system has been developed. Here are some specific benefits of using an automatic question tagging system:

- Enhanced searchability: Users can more easily locate the information they're looking for when questions are tagged with pertinent keywords since search filters allow users to focus their results.
- Faster response times: Instead of manually reviewing and classifying each query, human moderators or customer service representatives can save time and effort by using automated tagging.
- Improved data analysis: By tagging questions, you can find trends and patterns in the questions that are asked, which can help you make better decisions and improve your products and services.

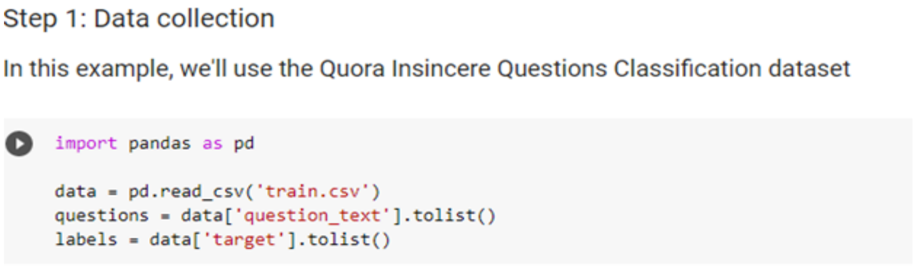
- Tagging can assist make sure that questions are sent to the right subject matter experts or knowledge articles, resulting in more accurate and beneficial responses.
- Better user experience: When the system can quickly and precisely identify the material that best fits users' requirements, users can get better support and answers to their inquiries.

Overall, by increasing the effectiveness and efficiency of question-and-answer systems, an automatic question-tagging system can offer considerable benefits for both users and companies.

5 Proposed Work

In our project, an automatic question tagging system typically involves the following steps:

- **Data Collection:** A large dataset of labeled questions is required to train and test the system. The dataset may be collected from various sources, such as question-and-answer websites, customer support forums, or educational platforms. In our project, we have used the Quora Insincere Questions Classification dataset.



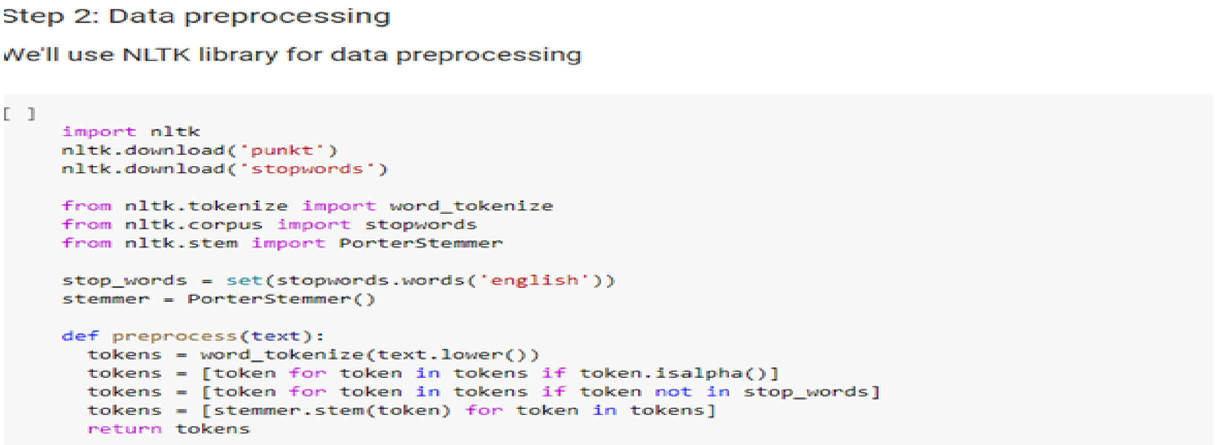
Step 1: Data collection

In this example, we'll use the Quora Insincere Questions Classification dataset

```
import pandas as pd

data = pd.read_csv('train.csv')
questions = data['question_text'].tolist()
labels = data['target'].tolist()
```

- **Preprocessing:** The dataset is preprocessed to clean and normalize the text, remove stop words, and extract features from the text, such as n-grams or part-of-speech tags.



Step 2: Data preprocessing

We'll use NLTK library for data preprocessing

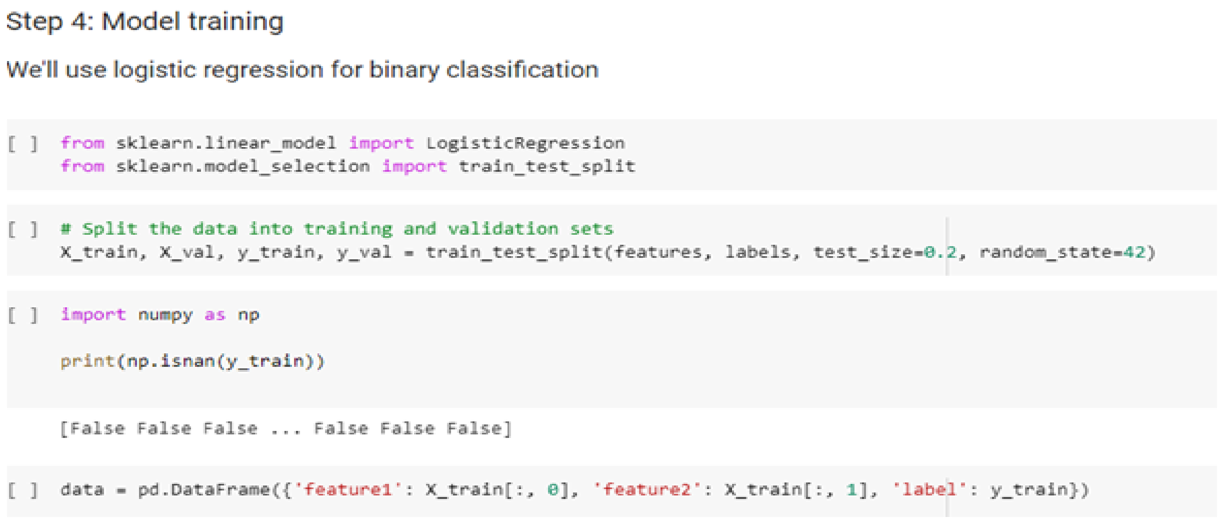
```
[ ]
import nltk
nltk.download('punkt')
nltk.download('stopwords')

from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer

stop_words = set(stopwords.words('english'))
stemmer = PorterStemmer()

def preprocess(text):
  tokens = word_tokenize(text.lower())
  tokens = [token for token in tokens if token.isalpha()]
  tokens = [token for token in tokens if token not in stop_words]
  tokens = [stemmer.stem(token) for token in tokens]
  return tokens
```

- **Training:** The preprocessed dataset is used to train a machine learning model, such as a decision tree, SVM, or neural network, to predict tags for new questions. The model is optimized using techniques such as cross-validation and hyperparameter tuning.



Step 4: Model training

We'll use logistic regression for binary classification

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

```
# Split the data into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(features, labels, test_size=0.2, random_state=42)
```

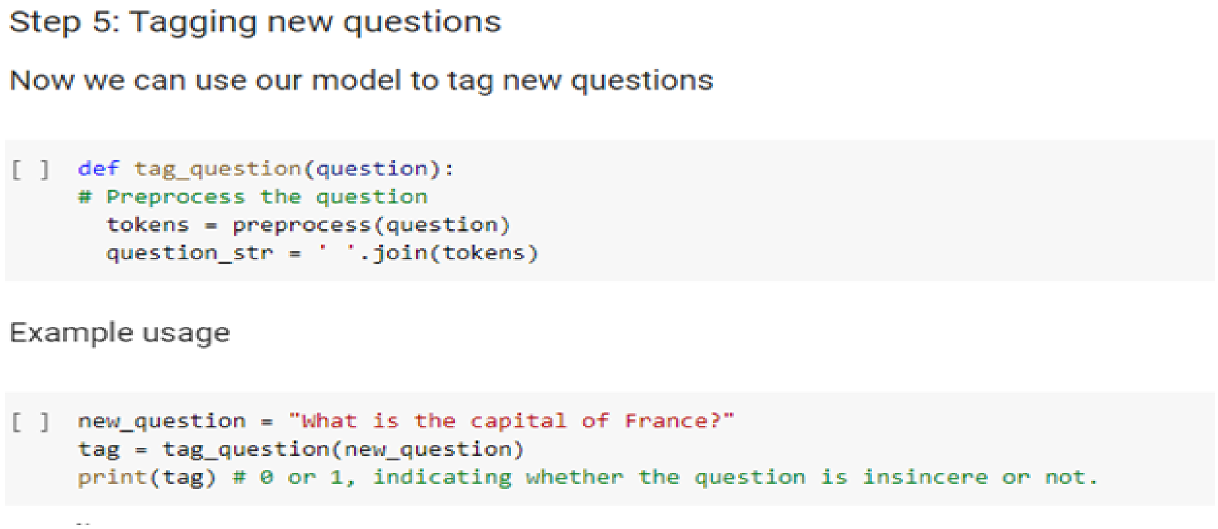
```
import numpy as np

print(np.isnan(y_train))
```

```
[False False False ... False False False]
```

```
data = pd.DataFrame({'feature1': X_train[:, 0], 'feature2': X_train[:, 1], 'label': y_train})
```

- **Tagging:** The trained model is used to automatically assign tags to new questions. The system may use a rule-based approach in conjunction with the machine learning model to improve accuracy.



Step 5: Tagging new questions

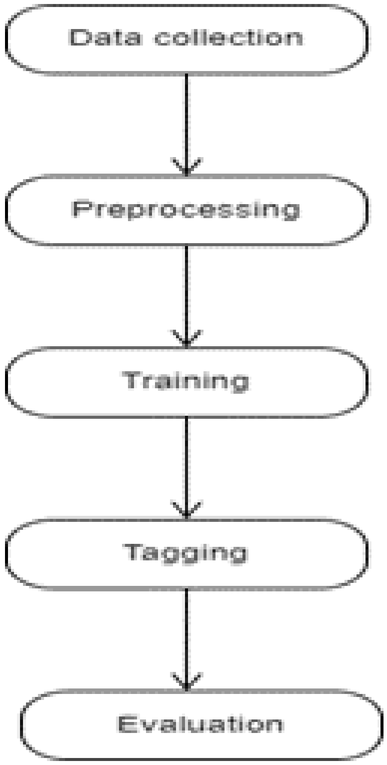
Now we can use our model to tag new questions

```
def tag_question(question):
# Preprocess the question
  tokens = preprocess(question)
  question_str = ' '.join(tokens)
```

Example usage

```
new_question = "What is the capital of France?"
tag = tag_question(new_question)
print(tag) # 0 or 1, indicating whether the question is insincere or not.
```

- **Evaluation:** The performance of the system is evaluated using metrics such as precision, recall, and F1 score. The system is refined and retrained as necessary to improve its accuracy and efficiency.



6 Evaluation Methodology

To ensure that an autonomous question tagging system is precise and efficient in its duty, evaluation is a crucial step in its development. Precision, recall, and F1 score are just a few of the evaluation measures that can be used to evaluate the system's performance.

How many of the system's tags are accurate is a measure of precision. It is calculated as the system's total number of tags divided by the number of proper tags assigned by the system.

The recall is a gauge of how many of the system's suggested tags are accurate for a particular inquiry. It is determined by dividing the system's assigned correct tags by the total number of accurate tags for the query.

F1 score is the harmonic mean of precision and recall and is often used as a combined measure of the system's performance. In addition to these indicators, it's crucial to take into account the automatic question tagging system's particular application and assess how well it works there.

For instance, in a customer support forum, comparing the tags that have been assigned to client queries to the actual issues or themes of the customer's query can be used to gauge how accurately the system tags customer questions.

Overall, the evaluation of an automatic question tagging system should be comprehensive and consider the metrics and factors relevant to its specific application.

7 Goal

The goal is to demonstrate that queries may be automatically tagged with pertinent terms in an accurate and timely manner and that doing so can make it easier for users to access the information they need. By automating the process of classifying and sorting questions, the study may also aim to show how the automatic tagging system might lessen the workload of human moderators or support staff.

The quality of the dataset, the performance of the rule-based system or machine learning model, and the applicability of the assessment measures will all have an impact on how well the study or experiment turns out in the end. If the study yields positive findings, more effective and practical automatic question tagging systems might be created, which would be advantageous for a variety of businesses and applications.