Automatic Question Tagging System

ABSTRACT:

Tagging is a popular approach to organizing information and searching content in information systems.
As a result, tags are used to categorize questions on sites like Quora and Stackoverflow. We have done our analysis on Kaggle's StackSample:10% of Stack Overflow Q&A. dataset. We have used "One-vs-Rest Classifier with "Support Vector Classification", "Logistic Regression", & "And Random Forest Classification ". We observed that Random Forest outperforms other algorithms with a Jaccard Score of 0.921 and Hamming Loss of 0.011.

INTRODUCTION:

Sites like Quora and Stackoverflow, which are especially created to have questions and answers for their users, frequently ask users to provide five words along with their questions so that they may be easily categorized. However, people occasionally offer incorrect tags, making it difficult for other users to explore. As a result, they want an automatic question tagging system that can recognize accurate and relevant tags for a user-submitted topic.

DATA:

Here is the link for the dataset: https://www.kaggle.com/stackoverflow/stacksample The text of 10% of the questions and answers from the Stack Overflow programming Q&A website is included in this dataset.

This is divided into 3 tables:

1. For all non-deleted Stack Overflow questions with an Id that is a multiple of 10, Questions provide the title, body, creation date, closed date (if applicable), score, and owner ID.
2. Each of the answers to these questions has a body, a creation date, a score, and an owner ID. The ParentId column references the Questions table.
3. Each question has its own set of tags, which are listed under Tags.

Data and Text Preprocessing:

1. Cleaning – Unwanted columns such as creation date, closed date, and score are removed.
2. Preprocessing of cleaned data such as "Tags", "Body", and "Title" is done (Removing HTML format, lowering text, transforming abbreviations, removing punctuations while popular tags like "C#" are taken care of, lemmatizing words, removing stop words).

Exploratory Data Analysis is done to find popular patterns and tags.

ANALYSIS AND RESULTS:

After classification and analysis are done, 80% of data is trained and used "One-vs-Rest Classifier with "Support Vector Classifier", "Logistic Regression", & "Random Forest Classification".

We are using the below evaluation metrics:

Jaccard Score – The Jaccard similarity index measures the similarity between two sets of data. It can range from 0 to 1. The higher the number, the more similar the two sets of data

Hamming Loss is the fraction of wrong labels to the total number of labels. In multi-label classification, hamming loss penalizes only the individual labels.

As we can observe the results of the models we used are as stated below:

Support Vector Classification:	Jaccard Score: 0.6473743535338152 Hamming loss: 0.0123506539413220
Logistic Regression:	Jaccard Score: 0.892213344137751 Hamming loss: 0.0124319547543301
Random Forest Classification:	Jaccard Score: 0.92153434145231 Hamming loss: 0.011539876329

Random Forest is the best fit model with the best Jaccard score and least Hamming loss.

CONCLUSION:

We have applied the ML algorithms to the TF-IDF vectorization of text rather than applying them to the title and body because it is giving the best accuracy compared to the vectorization of the body and title individually.

As we can observe, the “Jaccard score” which gives the similarity between the models is “high” and “Hamming loss” which provides data loss is “low” for Random Forest Classifier algorithm compared to the logistic regression classifier and Support Vector classifier. So, we have considered the Random Forest classifier as the best fit model for future analysis with reference to the dataset used.

FUTURE RESEARCH DIRECTIONS:

In the future, we can use the Artificial Neural Network which includes state-of-the-art NLP transformers such as BERT, GPT, etc so that we can retain the sequence information and feed the textual data directly to sequence-to-sequence models as stated in Appendix_1. In the traditional approaches as we have seen we first break down the sentence into tokens and then convert them to their numerical representations by calculating their frequencies or using the term frequency-inverse document frequency(TF-IDF approach).

MILESTONES AND REFERENCES:

Milestones - Data & Text Preprocessing, Basic Data Analysis on Tags, Supervised ML models. Reference:

1. https://en.wikipedia.org/wiki/Multi-label_classification#Statistics_and_evaluation_metrics
2. https://en.wikipedia.org/wiki/Multi-label_classification#Statistics_and_evaluation_metrics
3. https://www.kaggle.com/vikashrajluhaniwal/multi-label-classification-for-tag-prediction

APPENDIX:

1. https://jalammar.github.io/illustrated-transformer/
2. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf