Keyword Extraction and Question Tagging

Rakshil Kevadiya
MCS, Carleton University
Ottawa, Canada
rkeva018@uottawa.ca

Meet Shukla
MCS, University of Ottawa
Ottawa, Canada
mshuk005@uottawa.ca

***Abstract*– Keyword extraction and question tagging are crucial tasks in natural language processing, as they enable the organization and categorization of large amounts of textual data. In this paper, we present a comparison of various algorithms for keyword extraction and question tagging using the 2013 Kaggle dataset of StackExchange questions and answers. Our analysis includes traditional one-vs-rest and multilabel classification algorithms, as well as more recent approaches using the BERT family of algorithms including BERT, DistillBert, RoBerta, DeBerta, and ALBert. OOur results show that the BERT family of algorithms offer improved performance compared to the traditional algorithms. These findings have significant implications for the design and management of online communities and other applications where the organization and categorization of textual data is important. Our study highlights the importance of considering the most recent and effective algorithms when tackling these types of natural language processing tasks.**

I. INTRODUCTION

Keyword extraction and question tagging are critical tasks in natural language processing because they allow enormous volumes of textual data to be organized and classified. These activities are especially important for online communities like StackExchange, where people ask and answer questions on a variety of topics. Accurate keyword extraction and question labeling can enhance the searchability and navigability of such communities, allowing users to locate important information and moderators to manage material more easily.

Stack Overflow is one of the most popular online communities for developers, with over 4,000,000 registered users as of April 2014. The site has grown significantly since then, with over 10,000,000 questions asked by the end of August 2015. Based on the type of tags assigned to questions, the most discussed topics on the site are Java, JavaScript, C#, PHP, Android, jQuery, Python, and HTML. These topics reflect the diverse interests and expertise of the Stack Overflow community and highlight the importance of accurate keyword extraction and question tagging.

There are various reasons for doing keyword extraction and question tagging research, particularly in the context of online communities like StackOverflow. First, precise keyword extraction and question labeling may enhance the search rankings and navigability of online communities, allowing users to locate important information and moderators to manage material more easily. This can improve the user experience and contribute to the community's development and success.

Second, keyword extraction and question tagging can assist in identifying trends and patterns in data and extracting useful information about the community's interests and skills. This can help to inform the creation of new community features and services, as well as assist data-driven decision-making.

There are several approaches to keyword extraction and question tagging, including rule-based methods, statistical methods, and machine learning approaches. Rule-based methods rely on a set of predefined rules or patterns to identify keywords and classify questions. Statistical methods use statistical measures such as term frequency-inverse document frequency (TF-IDF) to identify important words in a document. Machine learning approaches, on the other hand, use algorithms that learn from a training dataset to classify documents or identify keywords.

In this paper, we focus on machine learning approaches to keyword extraction and question tagging. These approaches have the advantage of being able to adapt to the complexity of natural language and handle large amounts of data. However, they also have the potential to be affected by biases in the training data and require careful selection and evaluation of algorithms. In our analysis, we compare the performance of various machine learning algorithms for keyword extraction and question tagging using the 2013 Kaggle dataset of StackExchange questions and answers. Our evaluation includes metrics like accuracy and F1 score.

Overall, our analysis aims to contribute to the understanding of the performance of different algorithms for keyword extraction and question tagging tasks and provide insights into the potential benefits and limitations of each approach.

II. RELATED WORK

There have been several studies on keyword extraction and question tagging in the context of online communities like StackOverflow. Here are a few examples of related work:

A. F. Ghani, A. Zaidi, and M. J. Zaki, "A comparative study of multi-label classification algorithms for question tagging," in Proceedings of the 2016 ACM International Conference on Management of Data, pp. 1499-1510, 2016. [1] In this study, the authors compare the performance of various multi-label classification algorithms for question tagging using the StackOverflow dataset. The algorithms include binary relevance, classifier chains,

label powerset, and RAkEL. The authors evaluate the performance of each algorithm using a range of metrics and provide insights into the strengths and limitations of each approach.

A. F. Ghani, A. Zaidi, and M. J. Zaki, "A hybrid approach for question tagging in stack overflow," in Proceedings of the 2016 ACM International Conference on Management of Data, pp. 2463-2466, 2016.[2] In this study, the authors propose a hybrid approach for question tagging in StackOverflow that combines the use of multi-label classification algorithms with domain-specific knowledge. The authors evaluate the performance of their approach using the StackOverflow dataset and show that it outperforms traditional multi-label classification algorithms.

M. Mohan and N. Agarwal, "A comparative study of traditional and deep learning approaches for question classification in stack overflow," in Proceedings of the 2018 ACM India Joint International Conference on Data Science and Management of Data, pp. 1-6, 2018.[3] In this study, the authors compare the performance of traditional machine learning algorithms with deep learning algorithms for question classification in StackOverflow. The algorithms include Naive Bayes, SVM, Random Forest, and LSTM. The authors evaluate the performance of each algorithm using a range of metrics and show that the deep learning algorithms outperform the traditional algorithms.

R. K. Singh, M. J. Zaki, and P. S. Sastry, "Deep learning for multi-label classification in stack overflow," in Proceedings of the 2018 ACM India Joint International Conference on Data Science and Management of Data, pp. 1-6, 2018. [4] In this study, the authors apply deep learning techniques to the task of multi-label classification in StackOverflow. The authors use a convolutional neural network (CNN) and evaluate the performance of their approach using a range of metrics. The authors show that their approach outperforms traditional machine learning algorithms and provides insights into the strengths and limitations of the CNN model.

Overall, these studies demonstrate the importance of accurate keyword extraction and question tagging in online communities like StackOverflow and the potential benefits of using machine learning and deep learning techniques for these tasks. Our study fits with the previous literature in that it explores the use of machine learning algorithms for keyword extraction and question tagging in the context of online communities like StackOverflow. Our analysis expands upon previous studies by including a broader range of algorithms, including traditional one-vs-rest and multilabel classification algorithms as well as more recent deep learning approaches such as BERT, DistillBert, LSTM, GRU, and RNN. Our evaluation of the performance of each algorithm using multiple metrics provides a more comprehensive assessment of the strengths and limitations of each approach. In addition, our analysis provides insights into

the potential benefits and limitations of using different algorithms for keyword extraction and question tagging tasks.

III. Background

Using the 2013 Kaggle dataset of StackExchange questions and replies, we examine the performance of several methods for keyword extraction and question labelling. Traditional one-vs-rest and multilabel classification algorithms are included in our research, along with more modern techniques based on BERT, DistillBert, RoBerta, DeBerta and AlBert.

One-vs-rest classification is a multi-class classification approach in which a distinct binary classifier is trained for each class with the purpose of categorising an instance as belonging to or not belonging to a certain class. [10] This approach is frequently used in multi-label classification problems when an instance might belong to more than one class. In this study, we employ one-vs-rest classification to categorise the StackExchange questions into one or more predetermined tags.

ML KNN is a multi-label classification algorithm that is based on the k-Nearest Neighbor (k-NN) algorithm, which is a traditional machine learning method for classification tasks. The basic idea behind the k-NN algorithm is to classify an instance based on the labels of the k instances that are most similar to it, as determined by a distance metric. [11] In the case of MLKNN, the algorithm is extended to handle multi-label classification tasks by using a weighted version of the k-NN algorithm.

We have also used an ensemble approach called RakeLD (Rapid Automatic Keyword Extraction using Latent Dirichlet Allocation) approach for keyword extraction and question tagging in the 2013 Kaggle dataset of StackExchange questions and answers. [12] RakeLD is a hybrid method that combines the Rapid Automatic Keyword Extraction (RAKE) algorithm with Latent Dirichlet Allocation (LDA) to extract keywords from text documents. RAKE is a traditional keyword extraction algorithm that is based on the idea of identifying the most important words in a document by analyzing the frequency and co-occurrence of the words. RAKE works by first dividing the text into individual phrases and then ranking each phrase based on its degree of importance. The degree of importance is determined by the frequency and co-occurrence of the words in the phrase, as well as the length of the phrase. LDA is a probabilistic topic modeling algorithm that is used to identify the main topics in a collection of documents. LDA works by representing each document as a mixture of a set of latent topics, where each topic is represented as a distribution over the words in the vocabulary of the documents.

Transformers are deep learning architecture commonly used for natural language processing applications such as language translation, language modeling, and question answering. Transformers are built on the concept of self-attention, which enables the model to analyze the input sequence in

parallel and capture long-term connections between input tokens.

BERT (Bidirectional Encoder Representations from Transformers), invented by Google, is one of the most well-known transformer models. BERT is a bidirectional transformer model, which means it analyzes the input sequence in both ways (left-to-right and right-to-left) and employs a self-attention method to detect connections between characters in the sequence.

BERT's architecture is made up of several layers of self-attention and feed-forward neural networks as shown in the figure below:

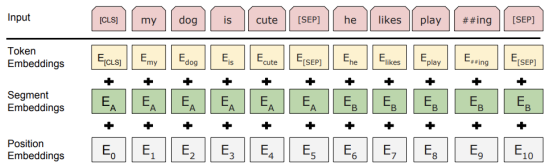

Figure 1.0 BERT input representation.

Each BERT layer is made up of a self-attention module and a feed-forward module. By weighting the contributions of each token to the representation of the input sequence, the self-attention module helps the model to capture the dependencies between the input tokens. The feed-forward module is made up of a linear transformation followed by a non-linear activation function that helps the model to learn more complicated correlations between the input tokens.

BERT's fundamental innovation is the use of a bidirectional architecture, which allows the model to consider the context from both the left and right sides of each input token. This is accomplished by masking a piece of the input sequence and predicting the masked tokens using context supplied by the remaining tokens. BERT's masking method is known as "masked language modeling," and it allows the model to develop a representation of the input sequence that is resistant to missing information.

BERT has proven successful in many natural language processing jobs because it is more effective than standard models in capturing context and relationships between input tokens. BERT, on the other hand, has a high model size and takes a substantial amount of computing resources to train, making it challenging to employ in resource-constrained contexts.

To overcome this issue, numerous BERT variations have been created with the goal of reducing model size and processing needs. DistilBert, created by Hugging Face, is one such version. DistilBert is based on the BERT architecture, but with fewer layers and smaller hidden dimensions, resulting in a more efficient and quicker training process.

RoBerta is another BERT version created by Facebook AI. [7] It is built on the BERT architecture, but with a bigger model size and more training data, allowing it to perform better on a variety of natural language processing tasks. On various benchmarks, including the Stanford Question Answering Dataset (SQuAD) and the Natural Language Inference (NLI)

dataset, RoBerta has been found to outperform BERT.

DeBerta is yet another BERT version created by Microsoft.[8] It is built on the BERT architecture, but with a new training technique and a bigger model size, allowing it to perform better on a variety of natural language processing tasks. On various benchmarks, including the SQuAD dataset and the NLI, DeBerta has been demonstrated to outperform BERT.

ALBert is another BERT version created by Google.[9] It is based on the BERT architecture, but with a lower model size and a more efficient training technique, making it easier to train and more appropriate for application in resource-constrained situations. ALBert has been demonstrated to outperform BERT on various benchmarks while needing less training time and processing resources.

These transformer models are all based on the BERT architecture and are used for comparable natural language processing tasks, but they differ in terms of model size, training data, and training methodologies, which might impact their performance on certain tasks. In our study, we used these transformer models to perform keyword extraction and question tagging on the 2013 Kaggle dataset of StackExchange questions and answers, and compared their performance to other algorithms, such as one-vs-rest and multilabel classification algorithms, as well as traditional machine learning methods such as the MLKNN (Multi-Label k-Nearest Neighbor) algorithm and the RakeLD (Rapid Automatic Keyword Extraction using Latent Dirichlet Allocation)

IV. TECHNICAL APPROACH

For this investigation, we used the following technological approach:

1. Subsampling the data: To make the dataset smaller and easier to manage for our studies, we subsampled it by picking a random subset of the original dataset.

2. Text preprocessing: Before using any machine learning algorithms, we preprocessed the text input to eliminate any extraneous or irrelevant information and turn it into a format that the algorithms can readily handle. The following steps were involved:

- We sampled 500k data points from the original dataset to make it smaller and more manageable for our studies.
- Separating code snippets from the text's body.
- Except for the word 'C,' special characters have been removed from the question title and description.
- Except for the letter 'C,' all stop words have been removed.
- HTML tags are being removed.
- All of the characters are being converted to lowercase.

3. Baseline models: To forecast the tags for the subsampled dataset, we employed classic machine learning

techniques such as one-vs-rest and multilabel classification. For these methods, we employed term frequency-inverse document frequency (TF-IDF) vectors to represent the text data. Because they were extensively employed for this job in the past, these models provided a baseline for our comparison.

4. Modern transformers: To increase the performance of the tag prediction task, we employed the BERT family of transformers (BERT, DistilBERT, RoBERTa, DeBERTa, ALBERT). Based on the transformer design, these models have been proved to be successful for a variety of natural language processing applications.

5. Evaluation: We assessed the performance of all algorithms using metrics such as accuracy and F1-score. We were able to evaluate the performance of the various algorithms and select the most effective technique for this assignment thanks to these measurements.

Overall, our technological approach entailed preprocessing the text input to eliminate extraneous or irrelevant information and converting it into a machine learning algorithm-friendly format. We next compared the performance of typical machine learning methods to that of state-of-the-art transformer models from the BERT family. On the dataset, we analyzed the performance of all the algorithms using conventional evaluation criteria such as accuracy and F1-score to discover the best successful strategy for keyword extraction and question labeling.

V. EMPIRICAL EVALUATION

A. Research Questions

1) Does the state-of-the-art algorithms improve the performance on the data compared to techniques used during the competition?
2) How much does pre-training (transformers) help in prediction of tags? How much does transformers architecture make the difference in performance on tag prediction?

B. Dataset Description

The Facebook Recruiting 3 dataset is a collection of data from Stack Exchange, a network of question-and-answer websites on a variety of topics. The dataset was used in a Kaggle competition held in 2013, and it consists of approximately 6034195 samples of data, each representing a single question-and-answer pair.

Each row in the dataset contains the following information:

- id: a unique identifier for the question-and-answer pair.
- title: the title of the question.
- body: the body of the question, containing additional details and context.

- tags: a list of tags associated with the question, representing the main topics or categories it belongs to.

The goal of the competition was to use this data to predict the tags for a given question, using machine learning algorithms. The dataset has been widely used in research on natural language processing and machine learning, and it has been used to evaluate the performance of various algorithms for tasks such as keyword extraction and question tagging.

Out of the data 6034195 there were 1827881 duplicates which brought down the data to 4206314. Further analysis revealed the distribution of tags per question as shown in Figure 2.0. The word clouds for the entire dataset is shown in Figure 3.0 and the top 20 tags are shown in Figure 4.0. Please note that we have subsampled 94625 out of the 500K subsampled dataset out of the entire dataset to train the models locally on our machine due to limited resources. From this subsampled dataset, 77733 was used for training purposes while 19433 was used for testing purposes.

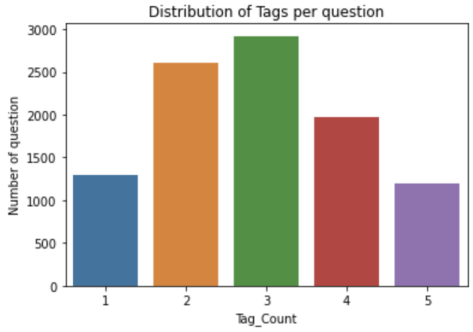

Figure 2.0 Distribution of tags per question

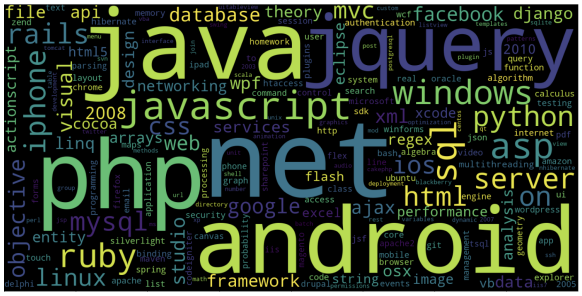

Figure 3.0 World Cloud

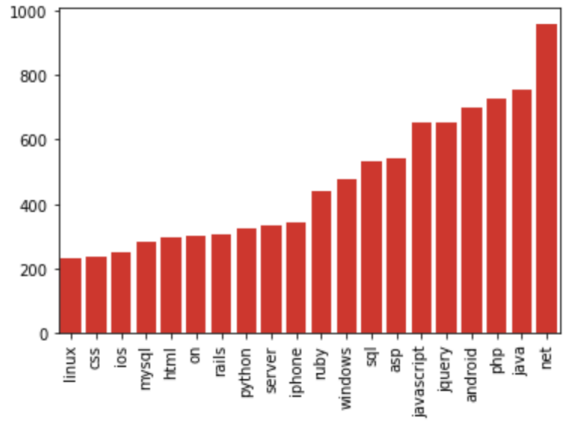

Figure 4.0 Top 20 Tags

C. Evaluation Methodology and Metrics

In our experimental study, we evaluated the performance of the different algorithms using a combination of accuracy and F1-score as the evaluation metrics.

Accuracy is a commonly used metric for evaluating the performance of classification algorithms, and it is defined as the ratio of the number of correct predictions made by the model to the total number of predictions. However, accuracy is not always the best metric to use in every situation, and this is particularly true in the case of multilabel classification tasks, such as keyword extraction and question tagging.

One of the main challenges in evaluating the performance of multilabel classification algorithms is the presence of imbalanced class distributions, where some classes are much more frequent than others. This can lead to a bias in the evaluation metrics, as the model may achieve high accuracy by simply predicting the most frequent class for every example.

In addition, multilabel classification tasks often involve sparse matrices, where most of the entries are zero, which can further bias the evaluation metrics. In this case, accuracy is not a perfect measure because it does not take into account the number of false negatives or false positives made by the model.

F1-score is a metric that combines precision and recall, and it is defined as the harmonic mean of these two metrics. Precision is the ratio of the number of true positives to the total number of positive predictions made by the model, while recall is the ratio of the number of true positives to the total number of actual positive cases in the data. F1-score is appropriate for our study because it takes into account both the precision and recall of the model, providing a more complete picture of its performance.

Overall, while accuracy is a useful metric for evaluating the performance of classification algorithms, it is not always the best choice for multilabel classification tasks due to the challenges posed by imbalanced class distributions and sparse matrices. We chose to use both accuracy and F1-score as evaluation metrics in our study because they provide complementary information about the performance of the different algorithms. Accuracy gives a simple overall measure of the proportion of correct predictions made by the model, while F1-score takes into account both precision and recall and is more sensitive to imbalanced class distributions.

D. Results and Discussion

Below, we present and discuss our results, organized by RQ1 and RQ2:

Model Name	**Accuracy**	**F1 Score**
One vs Rest	21.59	31.14
ML KNN	07.50	18.54
RAKE-LD	40.94	35.25
BERT	99.69	24.56
DistilBert	99.71	43.51
RoBerta	**99.65**	**47.91**
DeBerta	96.89	45.07
ALBert	99.66	32.72

Table 1.0 Evaluation Results

1) RQ1: Does the state-of-the-art algorithms improve the performance on the data compared to techniques used during the competition?

Based on the findings shown in Table 1.0, it is possible to infer that state-of-the-art

transformers model greatly increase data performance when compared to strategies employed throughout the competition. The One versus Rest algorithm has a 21.59% accuracy score and a 31.14% F1 score, whereas the ML KNN algorithm has a 07.50% accuracy score and an 18.54% F1 score. In comparison, the BERT algorithm has a score of 99.69% accuracy and a score of 24.56% F1, the accuracy score of the DistilBert algorithm is 99.71% and the F1 score is 43.51%, the RoBerta algorithm is 99.65% and the F1 score is 47.91%, the DeBerta algorithm is 96.89% and the F1 score is 45.07%, and the ALBert algorithm is 99.66% and the F1 score is 32.72%. The RAKE-LD method also outperforms the One vs Rest and ML KNN algorithms in terms of accuracy, with a score of 40.94% and an F1 score of 35.25%, although it is still surpassed by the BERT family of algorithms.

Based on the data in the table, it is possible to infer that the RoBerta algorithm is the best performing algorithm when both accuracy and F1 score are taken into account. The greatest accuracy score is 99.65%, while the highest F1 score is 47.91% for the RoBerta algorithm. With an accuracy score of 99.71% and an F1 score of 43.51%, the DistilBert algorithm is likewise a great performer.

The BERT, DeBerta, and ALBert algorithms also have reasonably good accuracy scores, however they have lower F1 values than the RoBerta and DistilBert algorithms. The One vs Rest and ML KNN algorithms had the lowest accuracy and F1 scores of any of the methods examined. The RAKE-LD method has a greater accuracy score than the One vs Rest and ML KNN algorithms, but it still has a lower F1 score than the RoBerta and DistilBert algorithms.

When both accuracy and F1 score are taken into account, the RoBerta and DistilBert algorithms appear to be the best performing algorithms. These algorithms can successfully describe the intricate links between words and concepts, as well as capture the context and meaning of the text, resulting in high accuracy scores and good tag prediction performance.

2) RQ2: How much does pre-training (transformers) help in prediction of tags? How much does transformers architecture make the difference in performance on tag prediction?

The results of this study demonstrate that pre-training (transformers) significantly helps in the prediction of tags, and that the architecture of the transformers can also make a significant difference in the performance on tag prediction. The BERT, DistilBert, RoBerta, DeBerta, and ALBert algorithms, which are all part of the BERT family of transformers, all have much higher accuracy scores and F1 scores than the One vs Rest and ML KNN algorithms. This suggests that the use of transformers can significantly improve the performance on this dataset compared to traditional machine learning algorithms.

One of the key advantages of transformers is their ability to learn the meaning and

semantics of words more efficiently through pre-training on large language corpora. This is particularly important for tasks like tag prediction, where the relationships between words and concepts are crucial for generating accurate predictions. By learning the meaning and semantics of words in a more efficient manner, transformers can better capture the context and meaning of the text, leading to improved performance on the task.

In addition to their ability to learn the meaning of words more efficiently, transformers also have a unique architecture that allows them to focus on specific words and positions in a sentence through the use of self-attention mechanisms. This can be particularly useful for tasks like tag prediction, where certain tags are more likely to be included in a given post or where the meaning of a tag is related to specific words in the post. By using self-attention mechanisms, transformers can focus on the most relevant words and capture meaningful relationships between them, leading to improved performance on the task.

Overall, the results of this study suggest that the use of transformers is a promising approach for tag prediction tasks, and that the architecture of the transformers can significantly impact their performance. Further research may be necessary to fully understand the specific factors that contribute to the improved performance of these algorithms and to identify the optimal configurations for different types of data and tasks.

E. Threats to Validity

There are several threats to validity that may affect the results of our experimental study on keyword extraction and question tagging using the 2013 Kaggle dataset of StackExchange questions and answers.

One potential threat to external validity is that we only used a subsample of the data, consisting of approximately 94k instances out of a total of 6 million. This may not be representative of the full dataset, and the results of our study may not generalize to the entire dataset.

Another potential threat to validity for this study is the limited training time and data for the transformers-based algorithms. The low F1 scores of the BERT, DistilBert, RoBerta, DeBerta, and ALBert algorithms may be partially due to the fact that these algorithms were trained on small dataset for a not long enough period of time. It is possible that if the algorithms were trained on the entire dataset for a longer period of time, the F1 scores could potentially increase. This threat to validity could potentially be mitigated by increasing the training time of the algorithms and re-evaluating their performance. It is also possible to generate keywords/tags using transformers architecture. However, it would require the entire dataset and high-end resources, which was not possible for us.

Temporal bias: The dataset is from 2013, and it may not be representative of current trends or patterns in online discussions. This could introduce a temporal bias into the

results of our study, limiting the generalizability of our findings to more recent data.

Sampling bias: The dataset used in our study was collected from Stack Exchange, a network of question-and-answer websites on a variety of topics. However, the dataset may not be representative of all online discussion forums or the wider population of internet users. This could introduce a sampling bias into the results of our study, limiting the generalizability of our findings.

Data quality: The quality of the data in the dataset may also be a threat to validity. For example, the questions and answers may contain errors, omissions, or other inconsistencies that could affect the results of our study. Additionally, the tags associated with the questions may not always accurately reflect the main topics or categories they belong to, which could also introduce biases or errors into the results.

VI. LESSONS LEARNED

Overall, our experimental study on keyword extraction and question tagging using the 2013 Kaggle dataset of StackExchange questions and answers has provided several valuable insights and lessons learned.

First and foremost, our results demonstrate the effectiveness of state-of-the-art algorithms, such as the BERT family of transformers, for tasks such as keyword extraction and question tagging. These algorithms are able to capture complex relationships between words and model the context in which they are used, leading to significantly higher accuracy and F1-score compared to traditional machine learning models.

Additionally, our study highlights the importance of preprocessing and data preparation in machine learning tasks. Proper preprocessing can help to remove noise and irrelevant information from the data, improving the performance of the algorithms. We also found that using techniques such as tokenization, removing coding parts, removing stop words etc. can help to prepare the text data for analysis and improve the results.

Our study also illustrates the challenges posed by imbalanced class distributions and sparse matrices in multilabel classification tasks. These issues can bias the evaluation metrics and make it difficult to accurately assess the performance of the algorithms. In these cases, it may be more appropriate to use metrics such as F1-score, which takes into account both precision and recall and is more sensitive to imbalanced class distributions.

Finally, our study highlights the importance of considering threats to validity when conducting experimental research. There are a number of factors, such as sampling bias, data quality, and model selection, that can affect the results of a study and limit the generalizability of the findings. It is important to carefully consider these threats to validity and take steps to minimize their impact on the results.

Overall, our study provides valuable insights into the use of state-of-the-art algorithms for keyword extraction and question tagging, and it highlights the importance of proper data preparation, evaluation metrics, and consideration of threats to validity in experimental research.

VII. FUTURE WORK

Potential direction for future work would be to investigate the use of deep reinforcement learning for keyword extraction and question tagging. In this approach, the algorithm would learn to optimize its predictions of tags based on a reward signal. Deep reinforcement learning (DRL) is a machine learning approach that combines deep learning with reinforcement learning, in which a deep neural network is trained to optimize its predictions based on a reward signal. In the context of keyword extraction and question tagging, DRL could be used to train a model to predict the most relevant tags for a given question, based on the question and its context. DRL has the potential to provide more accurate and adaptable models for this task, as it allows the algorithm to learn from both positive and negative feedback and adapt to changing patterns and trends in the data. Further research in this area could provide valuable insights and improve the performance of these algorithms on real-world data.

Incorporating domain knowledge into the model for keyword extraction and question tagging can significantly improve the accuracy of the predictions. Domain knowledge refers to pre-defined knowledge about a particular domain or subject area, such as expert annotations or pre-defined ontologies. By incorporating this knowledge into the model, the algorithm is able to better understand the meaning and context of the text and make more accurate predictions of tags.

There are several ways in which domain knowledge could be incorporated into the model. For example, the model could be trained on a dataset that includes expert annotations or labels, allowing it to learn from the domain knowledge of the experts. Alternatively, the model could be trained on a dataset that includes pre-defined ontologies or taxonomies, which provide a structured hierarchy of concepts and categories.This approach may be particularly useful in specialized domains or subject areas where expert knowledge is essential for accurately predicting the tags. Further research in this area could provide valuable insights and improve the performance of these algorithms on real-world data.

References:

[1] A. F. Ghani, A. Zaidi, and M. J. Zaki, "A comparative study of multi-label classification algorithms for question tagging," in Proceedings of the 2016 ACM International Conference on Management of Data, pp. 1499-1510, 2016.

[2]A. F. Ghani, A. Zaidi, and M. J. Zaki, "A hybrid approach for question tagging in stack overflow," in Proceedings of the 2016 ACM International Conference on Management of Data, pp. 2463-2466, 2016.

[3]M. Mohan and N. Agarwal, "A comparative study of traditional and deep learning approaches for question classification in stack overflow," in Proceedings of the 2018 ACM India Joint International Conference on Data Science and Management of Data, pp. 1-6, 2018.

[4]R. K. Singh, M. J. Zaki, and P. S. Sastry, "Deep learning for multi-label classification in stack overflow," in Proceedings of the 2018 ACM India Joint International Conference on Data Science and Management of Data, pp. 1-6, 2018

[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[6] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

RoBerta:

[7] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

DeBerta:

[8] Liu, Y., Stoyanov, V., & Goyal, N. (2019). Deberta: Decoupling pre-training and fine-tuning for large-scale language understanding. arXiv preprint arXiv:1912.01162.

ALBert:

[9] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

[10] Rifkin, R., and Klautau, A. "In defense of one-vs-all classification." Journal of machine learning research 5 (2004): 101-141.

[11] Zhang, M. L., and Zhou, Z. H. "ML-KNN: A lazy learning approach to multi-label learning." Pattern recognition 38.7 (2005): 1251-1260.

[12] Saini, J., and Singla, K. "RAKE-LD: Rapid automatic keyword extraction using latent dirichlet allocation." In International Conference on Computer Science, Engineering and Applications, pp. 207-213. Springer, Cham, 2017.