applied sciences

MDPI

Article

Unlocking the Potential of Keyword Extraction: The Need for Access to High-Quality Datasets

Zaira Hassan Amur [1,*], Yew Kwang Hooi [1], Gul Muhammad Soomro [2], Hina Bhanbhro [1], Said Karyem [2] and Najamudin Sohu [3]

[1] Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar 32160, Malaysia
[2] Faculty of Applied Informatics, Tomas Bata University, 760 01 Zlin, Czech Republic; krayem@utb.cz (S.K.)
[3] Department of Information Technology, Government College University, Hyderabad 17000, Pakistan
* Correspondence: zaira_20001009@utp.edu.my; Tel.: +60-148142485

Abstract: Keyword extraction is a critical task that enables various applications, including text classification, sentiment analysis, and information retrieval. However, the lack of a suitable dataset for semantic analysis of keyword extraction remains a serious problem that hinders progress in this field. Although some datasets exist for this task, they may not be representative, diverse, or of high quality, leading to suboptimal performance, inaccurate results, and reduced efficiency. To address this issue, we conducted a study to identify a suitable dataset for keyword extraction based on three key factors: dataset structure, complexity, and quality. The structure of a dataset should contain real-time data that is easily accessible and readable. The complexity should also reflect the diversity of sentences and their distribution in real-world scenarios. Finally, the quality of the dataset is a crucial factor in selecting a suitable dataset for keyword extraction. The quality depends on its accuracy, consistency, and completeness. The dataset should be annotated with high-quality labels that accurately reflect the keywords in the text. It should also be complete, with enough examples to accurately evaluate the performance of keyword extraction algorithms. Consistency in annotations is also essential, ensuring that the dataset is reliable and useful for further research.

Keywords: keyword extraction; natural language processing; dataset; structure; quality; complexity

Citation: Amur, Z.H.; Hooi, Y.K.; Soomro, G.M.; Bhanbhro, H.; Karyem, S.; Sohu, N. Unlocking the Potential of Keyword Extraction: The Need for Access to High-Quality Datasets. *Appl. Sci.* **2023**, *13*, 7228. https://doi.org/10.3390/app13127228

Academic Editor: Vincent A. Cicirello

Received: 20 March 2023
Revised: 10 May 2023
Accepted: 10 May 2023
Published: 16 June 2023

1. Introduction

Although there has been a significant advancement in recent years, the challenge of extracting significant keywords remains unresolved. Current algorithms for keyword extraction are not as efficient as those in many other fundamental domains of computer science, indicating that there is still room for improvement. The majority of established methods of deep learning typically involve a supervised methodology that is reliant on the availability of annotated text corpora for effective implementation. One of the initial methods for keyword extraction was presented by Turney [1], who created a specialized algorithm known as GenEx. Supervised techniques, such as Naïve Bayes, have been the primary method for identifying pertinent keywords in the majority of the approaches developed for keyword extraction. KEA [2] is perhaps the most widely used implementation of this approach, utilizing the Naïve Bayes machine learning algorithm for extracting keywords. Supervised methods, which are frequently more successful, suffer from the main limitation of having a relatively long training period. In contrast, unsupervised algorithms do not have the same training time constraints as supervised methods [3–5]. Unsupervised algorithms can easily be applied to documents in different languages or domains with little effort, as they are plug-and-play and can be executed quickly. However, unsupervised learning does not require labeled data. Instead, they use statistical patterns and relationships in the data. For example, in keyword extraction, unsupervised algorithms analyze the frequency and co-occurrence of words in a text corpus to identify

the most significant terms [3–5]. However, to achieve reliable and complete results, the quality of the input data is critical. If the data is noisy, incomplete, or inconsistent, the unsupervised algorithms may produce inaccurate or irrelevant results. For example, if the text contains spelling errors or non-standard abbreviations, the algorithm may miss important keywords or include irrelevant ones. Similarly, if the text is missing important context or background information, the algorithm may not be able to identify the most relevant keywords. Therefore, it needs to identify such data that is complete, error-free, and produces accurate results for the algorithms. Moreover, the availability of a high-quality, diverse dataset for keyword extraction could have a significant impact on the development of machine learning algorithms and natural language processing techniques.

Improved accuracy: A diverse dataset with high-quality data would allow researchers to develop more accurate machine-learning algorithms for keyword extraction. With access to a larger variety of text types and language structures, models could be trained to recognize patterns and relationships between keywords and text more effectively.

Generalizability: A diverse dataset would enable the development of keyword extraction models that are more generalizable across different domains and languages. This would be especially important for real-world applications, where models must be able to handle a wide range of text types and contexts.

Better evaluation: With access to a high-quality, diverse dataset, researchers could more effectively evaluate the performance of different keyword extraction models. This would make it easier to identify which models are most effective and pinpoint areas for improvement.

Increased innovation: A high-quality, diverse dataset would provide a foundation for new innovation in keyword extraction. Researchers would be able to build on existing work and develop new techniques and approaches for more effective semantic analysis.

1.1. Contribution of This Study

- Recognize the absence of a suitable dataset for keyword extraction.
- Emphasize the negative impact of using suboptimal datasets on efficiency and results.
- Conduct a study to identify a suitable dataset based on quality, complexity, and structure for semantic analysis of keyword extraction.
- Ensure the dataset is diverse and representative to capture variations in natural language processing.
- Create a pre-processing pipeline for semantic analysis of keyword extraction.
- Establish a canonical relationship between the dataset and the keyword extraction method.
- Promote the use of suitable datasets in future research.

1.2. Organization of the Study

The study is organized as follows: Section 2, discusses the methods of keyword extraction. Section 3 explains and proposes the available and suitable dataset for keyword extraction. Section 4 proposes the pre-processing pipeline for cleaning the text dataset for keyword extraction. Section 5 briefly discusses the whole study, and finally, the conclusion and future work are presented in Section 6.

2. Literature Review

The literature review focuses on the three current methods for keyword extraction. KeyBERT, YAKE, and RAKE.

2.1. KeyBERT

KeyBERT is a keyword extraction method in the Python library developed through research and development led by Mararten Grootendors. The model enables users to extract keywords or key phrases from the given text and embed sentences or documents into high-dimensional vector representations using BERT. KeyBERT builds on top of the hugging face library, which provides a user-friendly environment for working with pre-trained

BERT models [6]. KeyBERT also uses the TF-IDF and Maximal Marginal Relevance (MMR), which can be used to extract the most relevant and diverse keywords from given texts. In addition, KeyBERT also provides sentence embedding capabilities, which allow users to transform sentences or documents into high-dimensional vector representations that can be used for text classification or clustering. Golchin et al. [7] proposed KeyBERT for domain adaptation. They extract the abstractive summaries through neural language models and use these summaries for keyword extraction in semantic analysis. Khan et al. [8] used the KeyBERT model to extract the keywords from documents and match them with author-assigned keywords. They compare their model with other keyword extraction methods and find that KeyBERT extracts better keywords than others. Kelebercová et al. [9] used the KeyBERT model to extract true and fake news related to COVID-19 and identified the keywords as false and true news. Lee et al. [10] proposed a KeyBERT model for extracting the keywords from medical and non-medical service guidance. Piskorski et al. [11] used the KeyBERT model with other methods, such as YAKE and RAKE, to extract the lightweight keywords from new articles in a multilingual setup. Figure 1 illustrates the concept of how KeyBERT works.

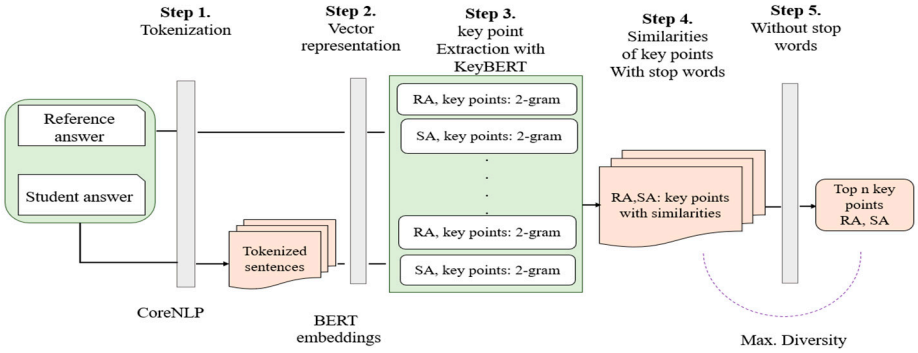

Figure 1. Keyword extraction from documents by utilizing the KeyBERT.

The procedure for extracting keywords from a document using KeyBERT involves installing the library, loading the target document, initializing the KeyBERT model with a pre-trained language model, calling the extract keywords function to obtain a list of top keywords along with their similarity score, and utilizing these keywords for various purposes such as improving search engine optimization, categorization, or identifying critical topics [11]. Overall, KeyBERT's keyword extraction process entails utilizing a pre-trained language model to identify the most relevant and informative words in a given document. To obtain representative keywords, the embeddings of each word and document are computed using Equation (1). Subsequently, the resulting embeddings are sorted in descending order, and the top n items are selected based on the highest similarity between the word and document embeddings. This similarity indicates the degree of representativeness of the document, with higher values indicating greater representativeness.

$$Similarity\ (A, B) = \frac{A \cdot B}{||A|| * ||B||} = \frac{\sum_{i=1}^{n} Ai * Bi}{\sqrt{\sum_{i=1}^{n} A_i^2} * \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

2.2. YAKE (YET Another Keyword Extractor)

YAKE is known as YET another Keyword Extractor! It is used to extract significant keywords from unstructured documents. A lightweight alternative method to unsupervised machine learning is to utilize local text features and statistical information, such as term frequencies and co-occurrences. This approach involves analyzing the document to identify common patterns and associations between terms without requiring pre-labeled training data [12]. By relying on statistical information and local text features, this method can provide a scalable and computationally efficient solution for keyword extraction tasks. Overall, this approach involves leveraging basic linguistic principles and statistical tech-

niques to extract the most relevant and informative keywords from individual documents. Campos et al. [13] used the YAKE model to extract multiple local features from a single document. The model is unable to understand the background information when extracting the words from the document. Meanwhile, it used the techniques of TF-IDF. Tohalino et al. [14] used YAKE to extract the keywords from the abstract and from the full paper to identify their relevance. They found it challenging to extract the keywords from a short text. Gadekar et al. [15] used the YAKE model to extract the keywords from web-based content; the keywords were further seeded to Guided Latent Dirichlet Allocation (GLDA) for classification purposes. Campos et al. [16] proposed the YAKE model to extract keywords from different text sizes. They also extract the word position and word size from the text. Following is Figure 2 presents an example of the YAKE algorithm extracting the keywords from the text. YAKE extracts the keywords based on the co-occurrences. However, they cannot be further utilized for contextual semantic analysis.

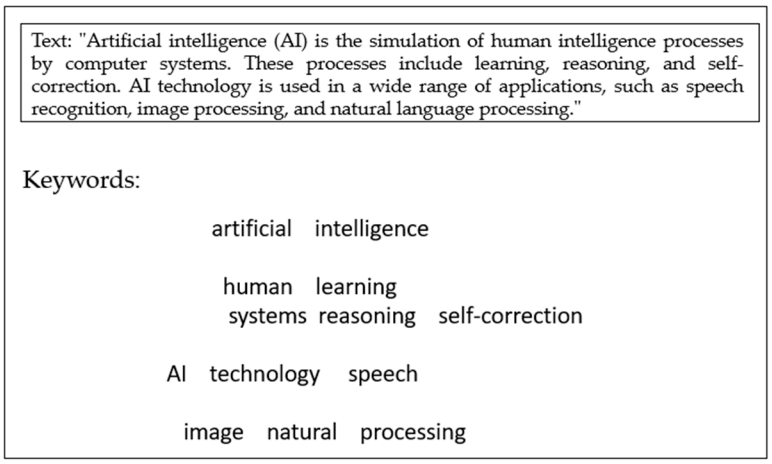

Figure 2. Example of the YAKE algorithm for extracting the keywords.

2.3. RAKE (Rapid Automatic Keyword Extraction)

RAKE is a graph-based algorithm that uses heuristics to extract keywords and key phrases from a text document. Rose et al. [17] conducted a study to compare the performance of RAKE with other keyword extraction algorithms on a dataset of research papers. They found that RAKE performed better than the other algorithms in terms of precision and recall. However, RAKE has some limitations, including the inability to distinguish between different parts of speech and the over-extraction of stop words. To address these limitations, Huang et al. [18] proposed an improved version of RAKE called NER-RAKE. Hu et al. [18] used the RAKE model to extract the keyword from patient information. They further used those keywords for patient classification. Thushara et al. [19] also used RAKE to extract the keywords from text summaries. Later, they compared the model with other keyword extraction models, such as TF-IDF and TextRank, and found that RAKE performed better than traditional approaches. Barun et al. [20] extracted the keywords from documents by using the RAKE model. Furthermore, they produce a candidate list of key phrases depending on the features of the word correlation. Figure 3 shows key phrase detection using the RAKE model. The RAKE model uses a parser to segment the text and apply segmentation. Tokenization is the main part, as it generates tokens for each word. After that, it applies the parts-of-speech tagging and extracts the most similar candidate key phrases from the text.

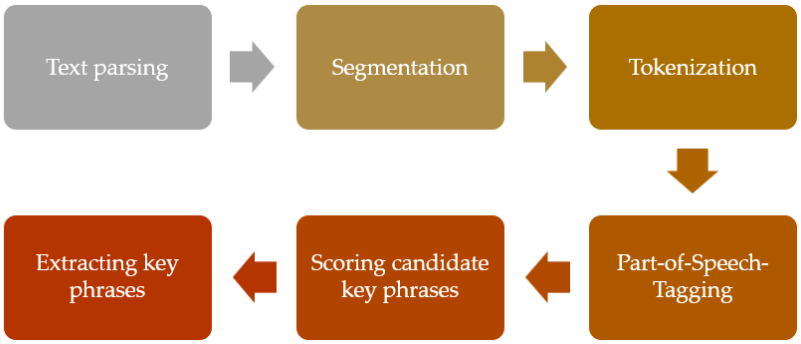

Figure 3. Key Phrase detection of RAKE Model.

2.4. Challenges in Keyword Extraction Methods (KeyBERT, YAKE, RAKE)

2.4.1. Challenges of the KeyBERT Model in Keyword Extraction

Availability and Quality of Data: As with any machine learning model, KeyBERT requires a substantial amount of data to learn effectively [21]. However, finding and curating a high-quality dataset that is representative of the target domain can be challenging. Poor-quality data, such as text that contains spelling or grammatical errors, can negatively impact the model's performance.

Domain-specificity: KeyBERT's performance can vary depending on the domain it is applied to. If a model is fine-tuned for one domain, it may not perform well on a different domain, as the keywords and context may differ [22,23]. Therefore, it is essential to have domain-specific datasets to ensure that the model performs well for the intended use case.

Data Pre-processing: Pre-processing data to make it suitable for the model can be time-consuming and require domain knowledge. Pre-processing includes cleaning the data, removing stop words, stemming, and lemmatizing, among other tasks. If the data is not pre-processed correctly, the model may miss or misclassify keywords.

Data Imbalance: The distribution of data can impact the performance of unsupervised models. If the dataset is imbalanced, meaning that some keywords occur much more frequently than others, the model may focus too much on the overrepresented keywords and overlook other important ones [24].

2.4.2. Challenges of the YAKE Model in Keyword Extraction

Ambiguity and Context Dependency: Text can be ambiguous and context-dependent, which can make it challenging to identify and extract keywords accurately [25,26]. Words can have multiple meanings, and the context in which they are used can change the interpretation of the text. Therefore, YAKE needs to be able to capture the context and semantics of the text accurately.

Length of Text: The length of the text can also affect the performance of YAKE. Shorter texts may not provide enough information to accurately identify keywords, while longer texts can be computationally expensive and require more processing power [27].

Data Quality: The quality of the input data can affect the performance of YAKE. Data that contains spelling or grammatical errors, incomplete sentences, or low-quality text may result in the extraction of irrelevant or inaccurate keywords [27,28].

Identification of Multi-word Phrases: YAKE may struggle with identifying multi-word phrases or compound words that are important keywords in some contexts. These phrases may not appear in a dictionary or stop-word list, making them difficult to identify using traditional methods [16].

2.4.3. Challenges of the RAKE Model in Keyword Extraction

Stop word Ambiguity: RAKE uses a list of stop words to identify and remove irrelevant words from the text. However, the definition of stop words can be ambiguous, and what

may be a stop word in one context may be an essential keyword in another context. This ambiguity can lead to the removal of critical keywords from the text [27–29].

Term Frequency-Inverse Document Frequency (TF-IDF) Sensitivity: RAKE uses TF-IDF to assign weights to keywords based on their frequency in the text and their frequency in the document corpus. However, the TF-IDF approach can be sensitive to outliers and may not accurately capture the importance of rare keywords [27].

Dependency on Punctuation: RAKE uses punctuation to identify phrase boundaries, making it sensitive to the punctuation used in the text [30]. Different types of punctuation may not be equally effective in identifying phrase boundaries, which can lead to errors in keyword extraction.

Inability to Handle Multi-word Phrases: RAKE may struggle with identifying multi-word phrases or compound words that are important keywords in some contexts. These phrases may not appear in a dictionary or stop-word list, making them difficult to identify using traditional methods [30].

Sensitivity to Text Length: RAKE's performance can be impacted by the length of the text. Shorter texts may not provide enough information to accurately identify keywords, while longer texts can be computationally expensive and require more processing power. Table 1 compares the challenges faced by the KeyBERT, YAKE, and RAKE models in keyword extraction across five different dimensions. The dimensions include domain-specificity, the need for high-quality data, ambiguity and context dependency, length of text, and difficulty in identifying multi-word phrases. The "✓" indicates that the model faces a particular challenge in that dimension. However, "x" indicates that the model is free from that challenge.

Table 1. Challenges faced by KeyBERT, YAKE, and RAKE.

Model	Domain Specificity	High-Quality Data	Context Dependency	Semantic Analysis	Multi-Word Phrases	Stop Word Ambiguity	TF-IDF Sensitivity	Dependency on Punctuations	Sensitivity to the Text Length
KeyBERT (Embedding based)	✓	✓	x	x	x	x	x	x	✓
YAKE (Rule-Based)	✓	✓	✓	✓	✓	x	✓	x	✓
RAKE (Statistical-based)	✓	x	x	✓	x	✓	✓	✓	✓

3. Dataset Identification for Keyword Extraction

Identifying a suitable dataset for keyword extraction depends on the specific task or application for which the keywords will be used. However, in general, a good dataset for keyword extraction should meet the following criteria:

- Large enough to cover a wide range of topics and domains.
- Diverse in terms of the types of documents included, such as news articles, academic papers, and social media posts.
- High quality, with accurate and well-formed text that is free from errors and inconsistencies.
- Annotated with ground-truth keywords that can be used for evaluating the performance of keyword extraction models.
- Includes a variety of text lengths, from short tweets to longer articles, to test the sensitivity of the model to text length.
- Representative of the target language or languages, with a variety of sentence structures.

3.1. Selection of Datasets for Keyword Extraction Analysis

In this section, we have assessed datasets based on their quality, structure, and complexity and have excluded certain datasets from further analysis due to not meeting our selection criteria. The datasets that have been excluded are as follows:

DUC2001: The Document Understanding Conference 2001 dataset is a commonly used benchmark dataset for keyword extraction, but it does not meet the complexity criteria because it only contains short news articles, which are relatively simple in terms of language and structure.

SemEval-2010 Task 5: This dataset is used for keyword extraction and classification, but it does not meet the quality criteria because it contains noisy text that is not easy to understand, such as spelling errors and non-standard language use [30,31].

KP20k: This dataset contains 20 million articles from various domains, but it does not meet the structure criteria because the articles are not annotated with clear keyword or key phrase labels, making it difficult to use for supervised learning, and unsupervised models require more time to process such information.

Open Directory Project: This dataset contains a large number of web pages categorized into various topics, but it does not meet the quality criteria because the web pages contain irrelevant or low-quality content, making it difficult to identify useful keywords.

We have selected the Twitter and Mohler datasets for keyword analysis due to their ease of implementation in unsupervised models. A specific amount of data was extracted from these datasets for keyword extraction, and the models used for experimentation, such as KeyBERT, RAKE, and YAKe, did not require any annotations.

3.1.1. Twitter Dataset

The Sentiment140 dataset: This dataset contains 1.6 million tweets that have been labeled as positive or negative based on the presence of positive or negative emotions in the tweet. The dataset has a balanced class distribution with 800,000 positive tweets and 800,000 negative tweets [31,32].

A Twitter dataset includes not only the text of tweets but also information about the sentiment of the tweets, such as whether they are positive, negative, or neutral. This is because Twitter is a popular platform for expressing opinions and emotions about various topics, such as products, politics, or entertainment. Twitter datasets that include sentiment information can be useful for keyword extraction tasks that require an understanding of the overall sentiment of the text. In the Twitter dataset, we can identify tweets related to a particular product, such as a new smartphone release, and use sentiment analysis techniques to classify them as positive, negative, or neutral based on the language used in the tweet. For instance, a tweet that says, "I love my new phone! The camera quality is amazing!" might be classified as positive, while a tweet that says "I'm so disappointed with the battery life of this new phone" might be classified as negative. Once we have classified the tweets by sentiment, we can use keyword extraction techniques to identify the most commonly used words or phrases in each sentiment category. For example, we might find that keywords such as "fast charging", "long battery life", and "great camera" are frequently associated with positive sentiment, while keywords such as "poor battery life" and "bad customer service", are associated with negative sentiment. This information could be useful for the company in improving their product or customer service by addressing the concerns and preferences of their customers, as reflected in the sentiment of their tweets. We have evaluated the datasets in terms of dataset structure, complexity, and quality [33], as depicted in Figure 4.

3.1.2. Challenges and Factors Considered while Analyzing the Twitter Dataset

Dataset Structure: The Twitter dataset is characterized by its unstructured and noisy nature. The dataset consists of tweets, which are short messages containing up to 280 characters. Each tweet may contain text and other metadata such as the author, timestamp, and location. The dataset may also include sentiment information, which can be positive, negative, or neutral. Some specific challenges that could arise in this context include:

Misspellings: The dataset contains spelling errors and non-standard spellings of words, which can lead to incorrect keyword extraction. For example, a tweet contains the word "awesum" instead of "awesome".

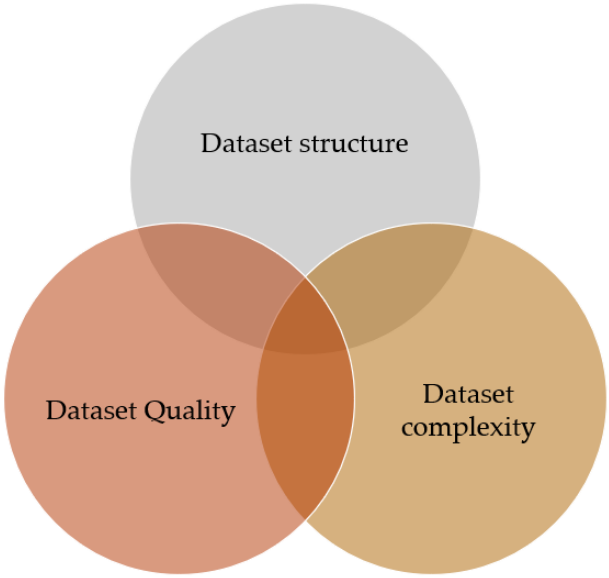

Figure 4. Dataset Structure, Complexity, and Quality.

Abbreviations: The dataset has many abbreviations to fit within the limited character count, which can make it difficult for keyword extraction models to correctly identify keywords. For example, a tweet might contain the abbreviation "LOL" instead of "laugh out loud".

Hashtags: The Twitter dataset uses hashtags to group tweets together by topic. While hashtags can be useful for keyword extraction, they can also be noisy and irrelevant. For example, a tweet might contain the hashtag #MondayMotivation, which may not be relevant to the content of the tweet.

Retweets: The dataset contains retweets of other users' tweets, which can lead to duplicate content and make it more difficult to identify unique keywords. For example, a tweet might be retweeted multiple times, leading to multiple identical instances of the same tweet in the dataset.

Readability and accessibility: The readability of the Twitter dataset depends on the specific context and research question. While some errors and challenges may arise, the dataset is generally accessible and can provide valuable insights into the sentiments and opinions of Twitter users on various topics.

Dataset Complexity: Lack of diversity in language: the dataset used similar language and expressions when discussing specific topics or events, making it challenging for keyword extraction tools to accurately identify and extract relevant keywords.

High noise-to-signal ratio: The dataset contains a lot of noise and irrelevant data. This can make it challenging for keyword extraction tools to filter out irrelevant data to extract relevant and meaningful keywords accurately.

Slang and jargon: The dataset uses slang and jargon that may not be well-known or understood by the general public. This can make it challenging to identify and extract keywords that are relevant to a broader audience.

Irony and sarcasm: The dataset uses irony and sarcasm in their tweets, which can be difficult for keyword extraction tools to detect. This can result in incorrect keyword extraction if the tool is not trained to understand the nuances of irony and sarcasm in Twitter language.

Ambiguity: The dataset is ambiguous and open to multiple interpretations. Keyword extraction tools must be fine-tuned to identify and extract the most likely interpretation of the tweet's content.

Dataset Quality: The quality of the Twitter dataset is influenced by factors such as the quality of the data collection process, the presence of spam or fake accounts, and the reliability of the sentiment analysis. Some specific challenges related to dataset quality include:

Bias in data collection: The dataset is biased towards certain types of users and topics, which can skew the dataset and affect the accuracy of keyword extraction models.

Diversity: The dataset contains different types of emotions, which presents the diverse nature of keyword extraction. We have found that there are some spam tweets that lack the consistency of the dataset. Table 2 provides a summary of the dataset's structure, complexity, and quality.

Table 2. The Twitter dataset in terms of Structure, Complexity, and Quality.

Aspect	Challenge	Example
Dataset Structure	Limited character count per tweet	"I love this product!" vs. "This product is terrible."
Dataset Structure	Use of non-standard language	"OMG this product is so lit 🔥👏."
Dataset Structure	Use of emoji emoticons	"I can't wait to see the new movie 🎬😍."
Dataset Complexity	Noisy and unstructured data	Tweets contain irrelevant information, such as hashtags and mentions.
Dataset Complexity	High volume of data	A large number of tweets are available; some of them are consistent and duplicates.
Dataset Quality	Bias in data collection	The dataset is skewed towards certain topics, leading to inaccurate keyword extraction results. However, it is balanced in terms of positive and negative tweets and is neutral.
Dataset Quality	Inconsistent labeling by annotators	Annotator 1 labels the following tweet as "negative": "I had a terrible experience with customer service today. They were unhelpful and rude." Annotator 2 labels the same tweet as "neutral": "I had an experience with customer service today. They were unhelpful and rude. Meanwhile, the dataset is biased towards positive and negative outcomes only.

3.1.3. Performance of KeyBERT, YAKE, and RAKE on the Twitter Dataset

From the Twitter dataset, we have applied three models to extract the keywords. These keywords can be further utilized for extracting information if they are positive or negative. Different senses can be applied to capture the label information. Following are the results of the KeyBERT model on the tweet: "I just love the new Avengers movie! #Marvel #AvengersEndgame".

In Figure 5, as we can see, the keywords "Avengers" and "Endgame" have the highest degree of similarity (0.474), which indicates that they are closely related in this context. Note: The values in the matrix represent the degree of similarity between each pair of keywords, with higher values indicating a stronger similarity. Moreover, the following are the results of YAKE on the same tweet in Figure 6.

In Figure 6, we can see that the keyword—Avengers—has a similarity score of one with itself, as expected. It also has some degree of similarity with the keywords "movie" and "new". Love: this keyword has a moderate level of similarity with the keyword "new". Movie: This keyword has a high level of similarity with the keywords "Avengers" and "Marvel". new: This keyword has some degree of similarity with the keywords "Avengers" and "love". #Endgame: this keyword has a high level of similarity with the keyword "Avengers". Marvel: this keyword has a high level of similarity with the keyword "movie" and a moderate level of similarity with the keyword "new". It is worth noting that the hashtags "#Endgame" and "#Marvel" have similarities with the keywords "Avengers" and "movie", respectively, but the model is unable to remove the hashtag. This can lead to inconsistencies in the similarity matrix. In light of this, the following Figure 7 presents the results of the RAKE model on the tweet.

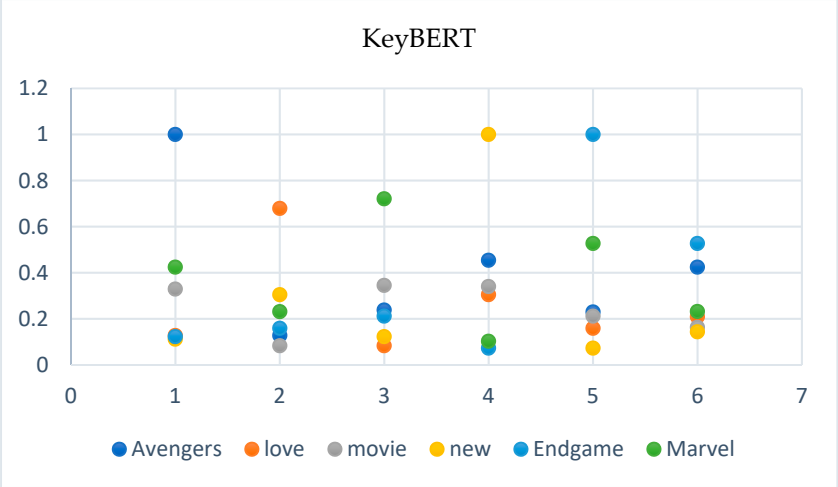

Figure 5. KeyBERT results on tweets.

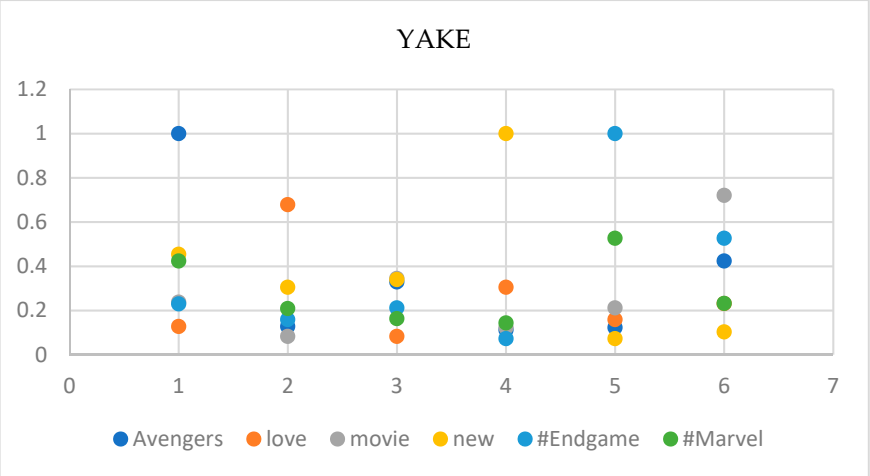

Figure 6. YAKE results on tweets.

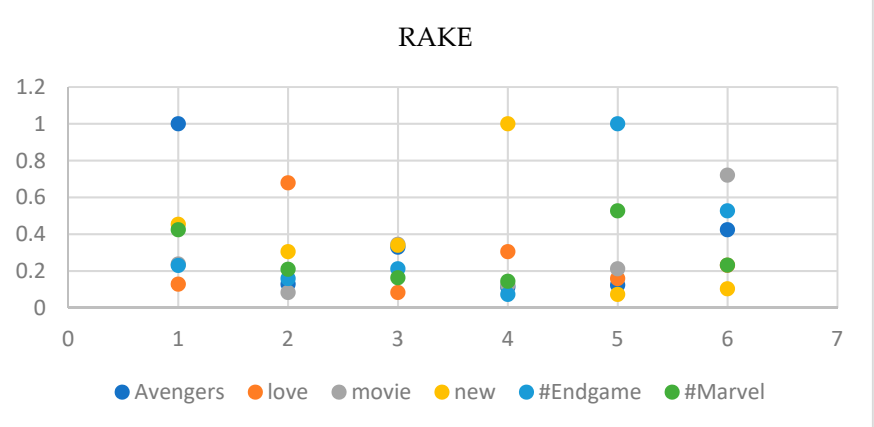

Figure 7. RAKE results on tweets.

The RAKE model identifies different keywords compared to KeyBERT and YAKE. It assigns different values to the similarity matrix, with some keywords having higher or lower similarities. For example, the RAKE model identifies the phrase "new Avengers

movie" as a single keyword with a high similarity score to the keyword "new". It also identifies the hashtag "#Endgame" as a separate keyword with a lower similarity score compared to YAKE. Overall, the RAKE model performs differently compared to KeyBERT and YAKE, showing that different keyword extraction methods can yield different results.

3.2. Mohler Dataset

This dataset is based on computer science questions and answers, particularly the Mohler Automated Student Assessment Grade (ASAG) dataset, which contains 2442 student responses to 87 questions on Data Structures from 12 assignments and two exams [34]. To evaluate each response, a numerical score between zero and five is based on a reference key provided by human evaluators. The final grade for each response is determined by taking the average of the scores given by two human graders. Several studies are still utilizing the Mohler Automated Student Assessment Grade (ASAG) dataset for different purposes, such as grading student responses, conducting semantic analysis, and selecting the best answers. Our objective is to evaluate the Mohler Automated Student Assessment Grade (ASAG) dataset in terms of its structure, complexity, and quality. This analysis will allow us to understand the characteristics of the dataset and assess its suitability for our research purposes. In terms of structure, we will examine the format and organization of the dataset, including its size, number of features, and data types. For complexity, we will evaluate the level of difficulty in processing the data and identifying patterns or relationships between variables. Finally, for quality, we will assess the accuracy and completeness of the data, as well as any potential biases or limitations that may affect its usability.

3.2.1. Dataset Structure

The structure of the Mohler Automated Student Assessment Grade (ASAG) dataset includes several components, as presented in Table 3. The dataset contains a unique ID for each record, which allows for easy tracking and referencing. The questions in the dataset are related to the domain of computer science, and for each question, there is a reference answer provided. The referenced answer serves as a standard solution for the question and can be used to evaluate the accuracy of student responses. The dataset also includes student answers, which are the actual responses provided by the students. These answers are of variable length and may contain relevant and irrelevant information. Each student's answer is evaluated by human graders, who assign a score ranging from zero to five to each response. This scoring system allows for the assessment of the quality and accuracy of each student's answer. Moreover, the dataset provides the average score given by two human evaluators for each student response, which serves as the final score for that particular response. This average score can be used as a basis for further analysis of the dataset.

Table 3. Mohler dataset structure.

Component	Description
Id	Unique identifier for each record.
Questions	Questions related to the computer science domain.
Reference answer	Standard solution for each question used to evaluate student responses.
Student answers	Actual responses provided by students, are of variable length and may contain relevant/irrelevant information.
Scores	Assigned to student answers by human graders on a scale of 0–5.
Average score	Given by two human evaluators for each student response, serves as the final score.

The dataset structure includes the following components that can aid in keyword extraction:

Questions: The questions in the dataset are related to the domain of computer science and can provide context and domain-specific terms for keyword extraction.

Reference answer: The referenced answer provides a standard solution for each question and can be used as a guide for identifying relevant keywords or key phrases.

Student answers: The actual responses provided by students can be mined for relevant keywords and phrases, which can help in assessing the quality and accuracy of the response.

Variable-length answers: The student answers in the dataset are of variable length, which may contain both relevant and irrelevant information. By extracting keywords from these responses, it is possible to identify the most important and relevant information [35].

ID: The unique ID for each record can be used to track and compare keywords and phrases across multiple student responses to the same question.

Accessibility: the dataset is publicly available and easily readable.

3.2.2. Dataset Complexity

The Mohler Automated Student Assessment Grade (ASAG) dataset has several complexities that should be considered when working with the data. These include:

Diversity in student answers: As the dataset includes responses from a large number of students, there is a significant amount of diversity in the answers provided. This can make it challenging to identify common keywords or phrases across all responses.

Variable-length answers: The length of student answers in the dataset is not consistent, which can complicate keyword extraction. Longer responses contain more relevant information, but they can also be more difficult to analyze.

Bias towards correct answers: The dataset is biased towards correct answers that are scored as five or less than five, as depicted in Figure 8b, as the reference answer is used as a guide for evaluating student responses. This can make it challenging to identify keywords or phrases related to incorrect or partially correct answers.

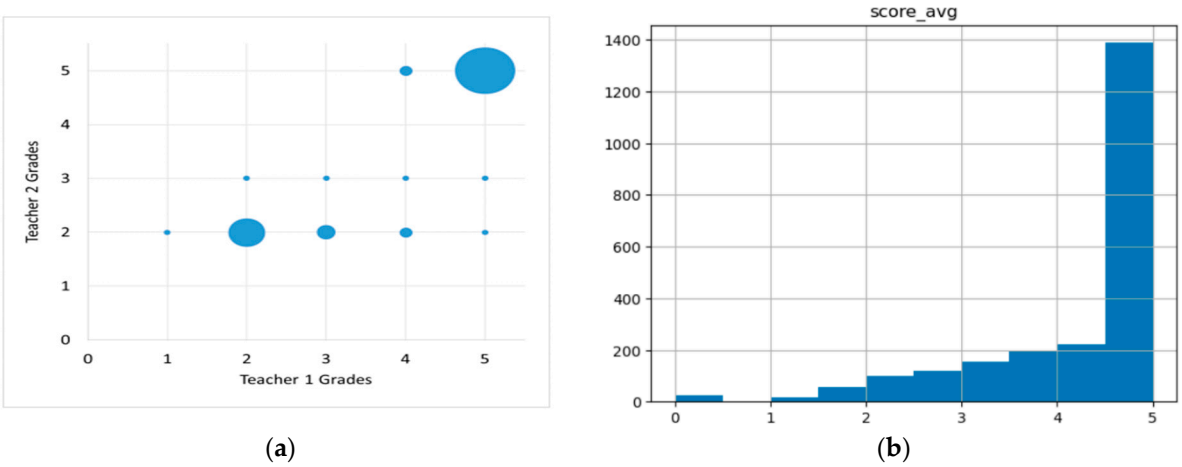

Figure 8. (**a**) Distribution of grades (left) and (**b**) dataset were biased towards correct answers (right).

Different scores assigned by human evaluators: The scores assigned by human evaluators may differ for different student answers, as depicted in Figure 8a, even when evaluating the same question. This can make it challenging to establish a consistent threshold for keyword extraction based on scores alone.

Despite these complexities, the Mohler ASAG dataset can still be useful for keyword extraction. To overcome these challenges, it may be necessary to use techniques such as natural language processing, machine learning, or crowdsourcing to identify common keywords or phrases across a large number of student responses. It may also be helpful to focus on identifying keywords and phrases related to specific concepts or topics within the domain of computer science rather than attempting to analyze the dataset as a whole.

3.2.3. Dataset Quality

The quality of the Mohler Automated Student Assessment Grade (ASAG) dataset has several factors that should be considered when working with the data. These include:

Spelling errors: The dataset contains several spelling errors, such as "levls" instead of "levels", "paramaters" instead of "parameters", "refrence" instead of "reference", "adress" instead of "address", "refinng" instead of "refining", "ponters" instead of "pointers", and "perenthesis" instead of "parenthesis". These errors can make it challenging to accurately assess and analyze student answers for keyword extraction.

Short answers: Some of the student answers in the dataset are too short to effectively assess for keyword extraction. This can make it difficult to identify relevant keywords or phrases that are unique to a particular question or topic.

Other quality factors: There may be other quality factors that affect the dataset, such as inconsistent formatting or grammar errors. These factors can make it challenging to accurately evaluate student answers and extract meaningful keywords or phrases.

To address these quality factors, it may be necessary to conduct pre-processing steps to clean the data before conducting keyword extraction. This can include techniques such as spell-checking, grammar-checking, and removing short answers. Additionally, it may be necessary to train machine learning models or use crowdsourcing techniques to improve the accuracy of keyword extraction from the dataset. The summary of the Mohler dataset is presented in Table 4.

Table 4. Summary of the Mohler dataset.

Factor	Description
Dataset structure	Contains ID, questions, reference answers, and student answers in variable lengths. Scored by human evaluators up to a scale of 0–5, an average score is also available.
Dataset complexity	Diversity in student answers, length is not consistent, the dataset is biased towards correct answers, and the scores of human evaluators are different on a number of answers.
Dataset Quality	Contains spelling errors such as "levls", "paramaters", "refrence", "adress", "refinng", "ponters", and "perenthesis". Some answers are too short to assess for keyword extraction. There may also be other quality factors that affect the dataset, such as inconsistent formatting or grammar errors.

3.3. KeyBERT, YAKE, and RAKE for Keyword Extraction on the Mohler Automated Student Assessment Grade (ASAG) Dataset

Table 5 shows the scores generated by YAKE, RAKE, and KeyBERT applied to the same reference answer "to simulate the behavior of portions of the desired software product".

Table 5. YAKE, RAKE, and KeyBERT on the Mohler dataset with one gram.

Keyword	Gram	YAKE Score	RAKE Score	KeyBERT Score
software	1	0.090	0.040	1.00
product	1	0.130	0.134	1.00
simulate	1	0.148	0.243	0.760
behavior	1	0.137	0.321	0.806
desired	1	0.106	0.110	0.765
portions	1	0.125	0.102	0.630

Table 5 displays the results of applying the YAKE, RAKE, and KeyBERT algorithms to extract keywords from a given text. The keywords are listed in the first column, and the second column indicates the number of words in each keyword (i.e., the gram). The third, fourth, and fifth columns show the scores assigned to each keyword by YAKE, RAKE,

and KeyBERT, respectively. In this particular example, KeyBERT provided the highest scores for all keywords, with a score of one assigned to "software product" and "product", indicating that these words are the most important and relevant in the given text. YAKE also assigned relatively high scores to the keywords "simulate", "behavior", "desired", and "portions", while RAKE assigned the lowest scores to all keywords. We again identify the most mixed results by extracting the two grams from the reference answers. We have noticed some slight changes. In Table 6, KeyBERT extracts the top five bi-grams with high scores. However, YAKE extracts the top five keywords but also generates duplicates such as "Desired", and "Desired portions" with a lower score than KeyBERT; in order to do this, RAKE was unable to remove the stop words and extract the top five bi-grams with stop words such as" the behavior", and "of portions" and generates a low score.

Table 6. KeyBERT, YAKE, and RAKE results with bi-gram.

Mohler Dataset					
KeyBERT		**YAKE**		**RAKE**	
Key Phrase (Bi-Gram)	**Score**	**Key Phrase (Bi-Gram)**	**Score**	**Key Phrase (Bi-Gram)**	**Score**
Software product	0.801	Software product	0.301	Software product	0.102
Simulate behavior	0.793	Simulate behavior	0.321	the behavior	0.493
Desired software	0.568	Desired	0.108	Desired software	0.234
Desired portions	0.743	Desired portions	0.240	of portions	0.323
Simulate product	0.835	Simulate product	0.435	product	0.165

3.4. Exploring the Suitability of KeyBERT, YAKE, and RAKE for Keyword Extraction across Datasets with Dataset Structure, Complexity, and Quality

Table 7 compares three different models for automated keyword extraction: KeyBERT, YAKE, and RAKE in terms of the dataset's structure, complexity, and quality. KeyBERT is best suited for datasets with a structured format and a standardized question–answer format. It works well on datasets with technical terms, domain-specific jargon, and high-quality context. YAKE works well on both structured and unstructured datasets but is best suited for datasets with domain-specific jargon [36,37]. However, it cannot identify technical terms or detect background information. RAKE also works well on both structured and unstructured datasets, but is best suited for datasets with straightforward language complexity. It can handle datasets with lower-quality data but is unable to remove noise from the text.

Table 7. Comparison of automated keyword extraction of KeyBERT, YAKE, and RAKE models.

Model	Dataset Structure	Dataset Complexity	Dataset Quality
KeyBERT	• Works well on structured datasets with a standardized question–answer format. • Has the potential to outperform YAKE and RAKE on structured datasets due to its ability to leverage context.	Works well on datasets with domain-specific jargon and technical terms.	Works well on datasets with high-quality context.
YAKE	• Works well on both structured and unstructured datasets.	Works well on datasets with domain-specific jargon but is unable to identify the technical terms.	Works well on an error-free dataset but is unable to detect background information.
RAKE	• Works well on both structured and unstructured datasets.	Works well on datasets with straightforward language complexity.	Works well on datasets with lower-quality data but is unable to remove the noise from the text.

3.5. Challenges and Limitations Faced by an Unsupervised Model on the Mohler Dataset

Short Answers: The dataset consists of very short answers, which poses challenges for keyword extraction methods. For instance, models such as RAKE and YAKE can only extract keywords based on term frequencies, while the KeyBERT model is limited in capturing the full context of the text due to the short nature of the sentences [38,39].

Biased: As mentioned earlier, the dataset is biased towards correct answers, which makes it difficult for keyword extraction methods to identify key concepts that do not align with the reference answers.

Inconsistent Evaluation: The answers in the dataset were graded by two human annotators on a zero–five scale. However, the grading can be inconsistent, as there may be variations in scores between the annotators for certain answers [40]. This inconsistency could pose difficulties for key phrase matching algorithms to compare with human annotators. One possible solution is to calculate the average score and compare the overall similarity score with that average score.

Limited Diversity: Another limitation of the dataset is its limited diversity in terms of domain and language. This could affect the generalizability of the keyword extraction models and their performance in real-world scenarios that involve diverse text types and structures.

4. Development of a Preprocessing Natural Language Pipeline for Cleaning the Text Dataset for Keyword Extraction

Above, Figure 9 shows the general preprocessing pipeline for keyword extraction for an unsupervised model, such as KeyBERT, YAKE, and RAKE. Following are the steps covered in the above figure:

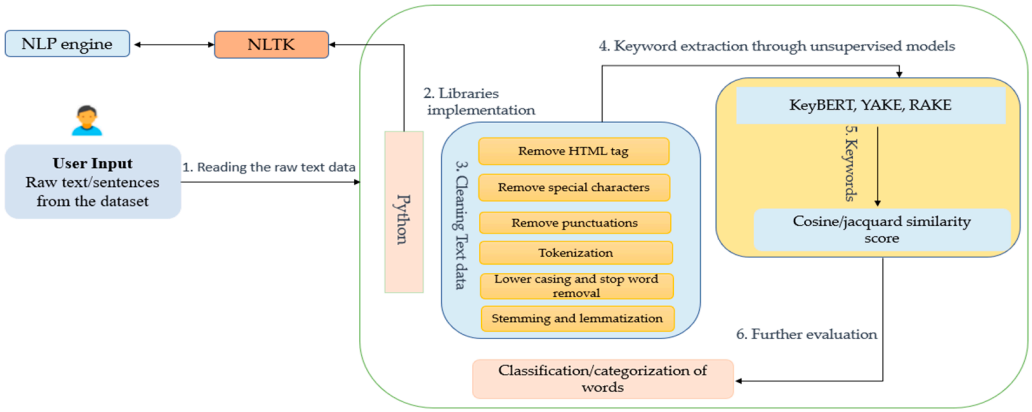

Figure 9. Preprocessing (NLP) pipeline for cleaning the text data.

Reading the raw text data: The first step in the pipeline would be to read the raw text data. This could be performed using a Python library such as Pandas or reading from a file. Meanwhile, from the NLP engine, the NLTK library is best suited for reading the text from the dataset.

Cleaning text dataset: To clean the text dataset, remove HTML tags. If the text data is scraped from a webpage, it may contain HTML tags that need to be removed before processing. Many text datasets contain contractions such as "can't" or "won't". These should be expanded to their full forms ("cannot", "will not") for consistency.

Removing special characters and punctuation: The next step would be to remove any special characters and punctuation marks from the text data.

Tokenization: The text data should be split into individual words or tokens for further processing.

Lowercasing and stop word removal: The text data should be converted to lowercase to ensure consistency, and common stop words such as "the" and "and" should be removed to reduce dimensionality.

Stemming or lemmatizing: The words in the text data should be stemmed or lemmatized to reduce redundancy and variation in word forms. Depending on the requirements of the project, part-of-speech tagging could also be used to identify specific types of words, such as nouns or verbs.

Keyword extraction using models: Finally, the preprocessed text data can be passed through the different keyword extraction models to generate a list of keywords. Here's how each of the models would fit into the pipeline:

KeyBERT: KeyBERT requires the input text to be in the form of a list of strings, so the preprocessed text data would need to be converted to this format before passing it to the model.

YAKE: YAKE can take in the preprocessed text data as a single string, so no additional formatting is required.

RAKE: RAKE requires the input text to be in the form of a single string, so the preprocessed text data would need to be concatenated into a single string before passing it to the model. After the keyword extraction, cosine similarity or jacquard similarity can be applied to identify the similar score. In Section 2, we have already discussed cosine similarity. However, Jaccard similarity is a measure of similarity between two sets of items, as mentioned in Equation (2). It is calculated by dividing the size of the intersection of the two sets by the size of the union of the two sets. The Jaccard similarity coefficient, J (A, B), can be defined as:

$$J(A, B) = \frac{|A \text{ n } B|}{|A \text{ u } B|}, \quad (2)$$

where A and B are sets of items, and |A| and |B| represent the number of items in each set. A Jaccard similarity coefficient of 1 indicates that the two sets are identical, while a coefficient of 0 indicates that the two sets have no items in common [41,42]. The jacquard similarity is commonly used in natural language processing and information retrieval tasks, such as measuring the similarity between documents or comparing sets of keywords. It can be a useful metric for identifying similarities between sets of items and can help researchers and practitioners make informed decisions about their projects.

Classification/categorization: After extracting the keywords, classification and categorization can be further used for evaluating the extracted keywords. In the case of sentiment analysis, an extracted entity can be classified or categorized as positive, negative, or neutral, and in the case of question–answering some matching rules can be applied to identify whether the entities reflect the correct or incorrect answers.

5. Discussion

This study has addressed the issue of inadequate datasets for keyword extraction methods. The absence of a suitable dataset could impede researchers from fully exploring the scope of semantic analysis for keyword extraction. It could also make it challenging to evaluate the performance of keyword extraction models, compare different approaches, and identify areas for improvement. Additionally, the unavailability of a suitable dataset could make it difficult to develop keyword extraction models that are relevant across diverse domains and use cases.

In this study, we have evaluated two datasets, Twitter and Mohler, based on their structure, complexity, and quality. This is an important step, as the quality of the dataset can greatly impact the performance of the models used for keyword extraction. By selecting a suitable dataset, a model can generate reliable and accurate results. After evaluating the datasets, we applied three different models, KeyBERT, YAKE, and RAKE, to extract keywords from the data. The results show that YAKE performs better on the Twitter dataset, whereas KeyBERT outperforms YAKE on the Mohler dataset. RAKE has a lower score compared to the other two models. This information can help researchers and practitioners select the most appropriate model for their specific dataset.

Furthermore, we have developed a general preprocessing pipeline for the three models to enhance the accuracy of keyword extraction. This is an important step, as preprocessing

can significantly impact the quality of the results. By developing a preprocessing pipeline, researchers can ensure that the models are working with clean and consistent data, leading to more accurate results.

In addition to evaluating the performance of the three models, KeyBERT, YAKE, and RAKE, we have also examined their limitations. This is an important aspect to consider, as it helps to understand the potential drawbacks of each model and the situations in which they may not perform optimally.

KeyBERT, for instance, relies heavily on pre-trained language models, which may not always capture the domain-specific context of the dataset being analyzed. This could lead to the extraction of less relevant keywords or the omission of important ones.

YAKE, on the other hand, relies on statistical properties of the dataset, such as term frequency and co-occurrence, which can be affected by noisy data or sparse datasets. This can result in the extraction of irrelevant or noisy keywords.

RAKE uses a simple and unsupervised approach to keyword extraction that relies on the extraction of frequent phrases from the text. However, this approach may not always capture the nuances of the text and may result in the extraction of uninformative phrases or stop words.

Overall, while the three models have shown promise in the context of research, it is important to keep their limitations in mind when selecting a model for a specific dataset or application. By understanding the limitations of each model, one can make an informed decision about which model to use and how to optimize its performance for the specific use case.

6. Conclusions

Our research has evaluated two datasets, Twitter and Mohler, based on their structure, complexity, and quality, and examined the performance of three different keyword extraction models, KeyBERT, YAKE, and RAKE. Based on our analysis, we have identified the strengths and limitations of each model and developed a general preprocessing pipeline to enhance their performance. Our research has significant implications for researchers and practitioners working on similar projects. By selecting an appropriate dataset and model and optimizing the preprocessing pipeline, they can achieve more accurate and reliable results. However, it is important to keep in mind the limitations of each model and select the one that is most suitable for the specific dataset and use case.

Moving forward, there are several areas for further research that could build upon the findings of this study. For example, exploring the performance of these models on different types of datasets, such as multilingual datasets, could provide valuable insights into their effectiveness in different contexts. Additionally, investigating the impact of different preprocessing techniques, such as stemming and lemmatization, could further enhance the accuracy and similarity of keyword extraction.

Overall, this research has provided valuable insights into the performance of different models for keyword extraction and highlighted the importance of dataset evaluation and preprocessing. By considering the strengths and limitations of each model, researchers and practitioners can make informed decisions and achieve more accurate and reliable results from their projects.

Author Contributions: Conceptualization, Z.H.A. and H.B.; Methodology, Z.H.A. and Y.K.H.; Validation, Z.H.A. and Y.K.H.; Formal analysis, Y.K.H. and H.B.; Investigation, Z.H.A., G.M.S. and H.B.; Data curation, Z.H.A. and H.B.; Writing—original draft, Z.H.A.; Visualization, Y.K.H., G.M.S., S.K. and N.S.; Supervision, Y.K.H.; Project administration, Y.K.H.; Funding acquisition, Y.K.H. All authors have read and agreed to the published version of the manuscript.

Funding: Cost Center 015PBC-005.

Informed Consent Statement: Not applicable.

Acknowledgments: Appreciation goes to the Pre-Commercialization-External: YUTP-PRG Cycle 2022 (015PBC-005).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Turney, P.D. Learning algorithms for keyphrase extraction. *Inf. Retr.* **2000**, *2*, 303–336. [CrossRef]
2. Witten, I.H. KEA: Practical automatic key phrase extraction. In Proceedings of the Fourth ACM Conference on Digital Libraries, Berkeley, CA, USA, 11–14 August 1999; pp. 254–255.
3. Rose, S.J.; Engel, D.; Cramer, N.; Cowley, W.E. *Automatic Keyword Extraction from Individual Documents*; Wiley: Hoboken, NJ, USA, 2010; pp. 1–20.
4. Wan, X.; Xiao, J. Single document keyphrase extraction using neighborhood knowledge. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, IL, USA, 13–17 July 2008; Volume 8, pp. 855–860.
5. Priyanshu, A.; Vijay, S.J. AdaptKeyBERT: An Attention-Based approach towards Few-Shot & Zero-Shot Domain Adaptation of KeyBERT. *arXiv* **2022**, arXiv:2211.07499.
6. Mihalcea, R.; Tarau, P. Textrank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 404–411.
7. Golchin, S.; Surdeanu, M.; Tavabi, N. A Compact Pretraining Approach for Neural Language Models. *arXiv* **2022**, arXiv:2208.12367.
8. Khan, M.Q.; Shahid, A.; Uddin, M.I.; Roman, M.; Alharbi, A.; Alosaimi, W.; Almalki, J.; Alshahrani, S.M. Impact analysis of keyword extraction using contextual word embedding. *PeerJ Comput. Sci.* **2022**, *8*, e967. [CrossRef] [PubMed]
9. Kelebercová, L.; Munk, M. Search queries related to COVID-19 based on keyword extraction. *Procedia Comput. Sci.* **2022**, *207*, 2618–2627. [CrossRef] [PubMed]
10. Lee, J.-S.; Hsiang, J. Patent classification by fine-tuning BERT language model. *World Pat. Inf.* **2020**, *61*, 101965. [CrossRef]
11. Surya, K.; Gayakwad, E.; Nallakaruppan, M. Deep learning for short answer scoring. *Int. J. Recent. Technol. Eng.* **2019**, *7*, 1712–1715.
12. Hu, Y.; Li, Y.; Yang, T.; Pan, Q. Short text classification with a convolutional neural networks based method. In Proceedings of the 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18–21 November 2018; pp. 1432–1435.
13. Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.; Nunes, C.; Jatowt, A. YAKE! Keyword extraction from single documents using multiple local features. *Inf. Sci.* **2020**, *509*, 257–289. [CrossRef]
14. Tohalino, J.A.; Silva, T.C.; Amancio, D.R. Using citation networks to evaluate the impact of text length on the identification of relevant concepts. *arXiv* **2023**, arXiv:2301.06168.
15. Gadekar, H.; Bugalia, N. YAKE-Guided LDA approach for automatic classification of construction safety reports. In Proceedings of the International Symposium on Automation and Robotics in Construction, Bogota, Colombia, 13–15 July 2022; pp. 451–458.
16. Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.M.; Nunes, C.; Jatowt, A. Yake! Collection-independent automatic keyword extractor. In *Advances in Information Retrieval, Proceedings of the 40th European Conference on IR Research, ECIR 2018, Grenoble, France, 26–29 March 2018*; Proceedings 40; Springer: Berlin/Heidelberg, Germany, 2018; pp. 806–810.
17. Sodhar, I.N.; Bhanbhro, H. Sindhi Language Processing on Online SindhiNLP Tool. *Univ. Sindh J. Inf. Commun. Technol.* **2020**, *4*, 4–7.
18. Hu, J.; Li, S.; Yao, Y.; Yu, L.; Yang, G.; Hu, J. Patent keyword extraction algorithm based on distributed representation for patent classification. *Entropy* **2018**, *20*, 104. [CrossRef] [PubMed]
19. Thushara, M.; Mownika, T.; Mangamuru, R. A comparative study on different keyword extraction algorithms. In Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 27–29 March 2019; pp. 969–973.
20. Baruni, J.S.; Sathiaseelan, J.G.R. Keyphrase Extraction from Document Using RAKE and TextRank Algorithms. *Int. J. Comput. Sci. Mob. Comput.* **2020**, *9*, 83–93. [CrossRef]
21. Amur, Z.H.; Hooi, Y.; Sodhar, I.N.; Bhanbhro, H.; Dahri, K. State-of-the Art: Short Text Semantic Similarity (STSS) Techniques in Question Answering Systems (QAS). In Proceedings of the International Conference on Artificial Intelligence for Smart Community: AISC 2020, Seri Iskandar, Malaysia, 17–18 December 2022; pp. 1033–1044.
22. Amur, Z.H.; Hooi, Y.K.; Soomro, G.M. Automatic Short Answer Grading (ASAG) using Attention-Based Deep Learning MODEL. In Proceedings of the 2022 International Conference on Digital Transformation and Intelligence (ICDI), Sarawak, Malaysia, 1–2 December 2022.
23. Amur, Z.H.; Hooi, Y.K. State-of-the-Art: Assessing Semantic Similarity in Automated Short-Answer Grading Systems. *Inf. Sci. Lett.* **2022**, *11*, 1851–1858.
24. Bhanbhro, H.; Hooi, Y.K.; Hassan, Z. Modern Approaches towards Object Detection of Complex Engineering Drawings. In Proceedings of the 2022 International Conference on Digital Transformation and Intelligence (ICDI), Sarawak, Malaysia, 1–2 December 2022.
25. Kadhim, A.I. Survey on supervised machine learning techniques for automatic text classification. *Artif. Intell. Rev.* **2019**, *52*, 273–292. [CrossRef]
26. Lyu, B.; Chen, L. Let: Linguistic knowledge enhanced graph transformer for chinese short text matching. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; pp. 13498–13506.

27. Miah, M.S.U.; Sulaiman, J.; Bin Sarwar, T.; Zamli, K.Z.; Jose, R. Study of keyword extraction techniques for electric double-layer capacitor domain using text similarity indexes: An experimental analysis. *Complexity* **2021**, *2021*, 8192320. [CrossRef]
28. Reategui, E.; Bigolin, M.; Carniato, M.; dos Santos, R.A. Evaluating the Performance of SOBEK Text Mining Keyword Extraction Algorithm. In Proceedings of the Machine Learning and Knowledge Extraction: 6th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2022, Vienna, Austria, 23–26 August 2022; pp. 233–243.
29. Tang, M.; Gandhi, P.; Kabir, M. Progress notes classification and keyword extraction using attention-based deep learning models with BERT. *arXiv* **2019**, arXiv:1910.05786.
30. Huang, H.; Wang, X.; Wang, H. NER-RAKE: An improved rapid automatic keyword extraction method for scientific literatures based on named entity recognition. *Proc. Assoc. Inf. Sci. Technol.* **2020**, *57*, e374. [CrossRef]
31. Imran, A.S.; Daudpota, S.M.; Kastrati, Z.; Bhatra, R. Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on COVID-19 related tweets. *IEEE Access* **2020**, *8*, 181074–181090. [CrossRef]
32. Dang, N.C.; Moreno-García, M.N.; De La Prieta, F. Sentiment analysis based on deep learning: A comparative study. *Electronics* **2020**, *9*, 483. [CrossRef]
33. Blake, R.; Mangiameli, P. The effects and interactions of data quality and problem complexity on classification. *J. Data Inf. Qual.* **2011**, *2*, 1–28. [CrossRef]
34. Mohler, M.; Bunescu, R.; Mihalcea, R. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 752–762.
35. Firoozeh, N.; Nazarenko, A.; Alizon, F.; Daille, B.J.N.L.E. Keyword extraction: Issues and methods. *Nat. Lang. Eng.* **2020**, *26*, 259–291. [CrossRef]
36. Fernando, B.; Herath, S. Anticipating human actions by correlating past with the future with Jaccard similarity measures. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13219–13228.
37. Alaggio, R.; Amador, C.; Anagnostopoulos, I.; Attygalle, A.D.; Araujo, I.B.D.O.; Berti, E.; Bhagat, G.; Borges, A.M.; Boyer, D.; Calaminici, M.; et al. The 5th edition of the World Health Organization Classification of Haematolymphoid Tumours: Lymphoid Neoplasms. *Leukemia* **2022**, *36*, 1720–1748. [CrossRef] [PubMed]
38. Huang, Z.; Xie, Z. A patent keywords extraction method using TextRank model with prior public knowledge. *Complex Intell. Syst.* **2021**, *8*, 1–12. [CrossRef]
39. Martinc, M.; Škrlj, B.; Pollak, S. TNT-KID: Transformer-based neural tagger for keyword identification. *Nat. Lang. Eng.* **2021**, *28*, 409–448. [CrossRef]
40. Jain, P.K.; Quamer, W.; Pamula, R.; Saravanan, V. Employing BERT-DCNN with semantic knowledge base for social media sentiment analysis. *J. Ambient. Intell. Humaniz. Comput.* **2022**. [CrossRef]
41. Amur, Z.H.; Hooi, Y.K.; Bhanbhro, H.; Dahri, K.; Soomro, G.M. Short-Text Semantic Similarity (STSS): Techniques, Challenges and Future Perspectives. *Appl. Sci.* **2023**, *13*, 3911. [CrossRef]
42. Gilal, A.R.; Waqas, A.; Talpur, B.A.; Abro, R.A.; Jaafar, J.; Amur, Z.H. In Question Guru: An Automated Multiple-Choice Question Generation System. In Proceedings of the 2nd International Conference on Emerging Technologies and Intelligent Systems: ICETIS 2022, Online, 2–3 September 2022; Volume 2, pp. 501–514.