PeerJ Computer Science

Evaluating keyphrase extraction algorithms for finding similar news articles using lexical similarity calculation and semantic relatedness measurement by word embedding

Talha Bin Sarwar, Noorhuzaimi Mohd Noor and M. Saef Ullah Miah

Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang,
Pekan, Pahang, Malaysia

Submitted 21 December 2021
Accepted 10 June 2022
Published 7 July 2022

Corresponding authors
Talha Bin Sarwar,
talhasarwar40@gmail.com
Noorhuzaimi Mohd Noor,
nhuzaimi@ump.edu.my

Academic editor
Imran Ashraf

Additional Information and Declarations can be found on page 21

DOI 10.7717/peerj-cs.1024

ABSTRACT

A textual data processing task that involves the automatic extraction of relevant and salient keyphrases from a document that expresses all the important concepts of the document is called keyphrase extraction. Due to technological advancements, the amount of textual information on the Internet is rapidly increasing as a lot of textual information is processed online in various domains such as offices, news portals, or for research purposes. Given the exponential increase of news articles on the Internet, manually searching for similar news articles by reading the entire news content that matches the user's interests has become a time-consuming and tedious task. Therefore, automatically finding similar news articles can be a significant task in text processing. In this context, keyphrase extraction algorithms can extract information from news articles. However, selecting the most appropriate algorithm is also a problem. Therefore, this study analyzes various supervised and unsupervised keyphrase extraction algorithms, namely KEA, KP-Miner, YAKE, MultipartiteRank, TopicRank, and TeKET, which are used to extract keyphrases from news articles. The extracted keyphrases are used to compute lexical and semantic similarity to find similar news articles. The lexical similarity is calculated using the Cosine and Jaccard similarity techniques. In addition, semantic similarity is calculated using a word embedding technique called Word2Vec in combination with the Cosine similarity measure. The experimental results show that the KP-Miner keyphrase extraction algorithm, together with the Cosine similarity calculation using Word2Vec (Cosine-Word2Vec), outperforms the other combinations of keyphrase extraction algorithms and similarity calculation techniques to find similar news articles. The similar articles identified using KPMiner and the Cosine similarity measure with Word2Vec appear to be relevant to a particular news article and thus show satisfactory performance with a Normalized Discounted Cumulative Gain (NDCG) value of 0.97. This study proposes a method for finding similar news articles that can be used in conjunction with other methods already in use.

Subjects Computational Linguistics, Data Mining and Machine Learning, Natural Language and Speech
Keywords Similar news article, Keyphrase extraction algorithm, Similarity calculation, Cosine similarity, Jaccard similarity, Word2Vec

How to cite this article Sarwar TB, Noor NM, Saef Ullah Miah M. 2022. Evaluating keyphrase extraction algorithms for finding similar news articles using lexical similarity calculation and semantic relatedness measurement by word embedding. **PeerJ Comput. Sci. 8:e1024 DOI 10.7717/peerj-cs.1024**

INTRODUCTION

In recent years, as a result of the exponential growth and development of information available through textual data and the Internet, finding and effectively managing relevant data has become a significant focus of academic research. Textual information can be either unstructured or semi-structured online; examples include online news and books, discussion forums, and academic papers. The challenges posed by online textual data have led to a variety of research initiatives in the areas of Information Retrieval (IR) and Natural Language Processing (NLP). Nowadays, Internet search engines facilitate the retrieval of relevant information by matching a user's keywords with a comprehensive database of extracted keywords from online text materials. Identifying and extracting the most important keywords that are useful and meaningful within the text is an essential part of dealing with textual materials, as the main themes of a large text or a single document can be characterized and captured using the extracted keywords or keyphrases (*Hasan & Ng, 2014*). Therefore, one of the most important research activities is to extract relevant keywords or keyphrases from a large textual material, and for this purpose text processing is a very crucial part (*Miah et al., 2022*). A keyphrase can be a word or a combination of words that define a specific and concise expression of one or more documents. Keyphrases convey the main idea of the document and help the reader decide whether to read further or look for additional details. They allow the reader to quickly decide if the article is the right one for them. Due to the growing amount of textual data, manual text processing and keyphrase retrieval is no longer feasible, which highlights the efforts to cope with the voluminous modern data by promoting the development of automated keyphrase extraction algorithms that leverage the massive processing resources of computers to replace manual work (*Babar & Patil, 2015*). As a result, automated keyphrase extraction has become a significant research interest in IR and text processing (*Welleck et al., 2019*).

Researchers have proposed several keyphrase extraction algorithms. These keyphrase extraction algorithms can be classified into two categories, namely supervised and unsupervised algorithms. In supervised algorithms, keyphrase extraction becomes a classification problem in which sentences are divided into keyphrase and non-keyphrase categories. Similar to other tasks involving supervised algorithms, a significant amount of domain-dependent labeled training data is required. The labeled *corpus* should be adjusted whenever the domain changes. Although labeling the *corpus* is a tedious and time-consuming process, the most traditional and popular supervised keyphrase extraction is KEA (*Witten et al., 1999*). There are also several algorithms for unsupervised keyphrase extraction. Based on the computational analysis, these algorithms can be classified into three categories, namely tree-based, graph-based, and statistical-based algorithms (*Rabby et al., 2020*). TeKET is the only tree-based algorithm that extracts high-quality keyphrases and performs well on research articles (*Rabby et al., 2020*; *Sarwar et al., 2021*). In addition to tree-based algorithms, several graph-based algorithms have also been proposed. Among the graph-based algorithms, MultipartiteRank (MR) (*Boudin, 2018*) and TopicRank (TR) (*Bougouin, Boudin & Daille, 2013*) are widely used (*Miah et al., 2021*;

Sarwar & Noor, 2021). On the other hand, among the statistical-based algorithms, KP-Miner (*El-Beltagy & Rafea, 2009*) and YAKE (*Campos et al., 2020*) are the most widely used algorithms (*Miah et al., 2021*).

The applications of keyphrase extraction can be varied, such as IR (*Azad & Deepak, 2019*), text summarization (*Zha, 2002*), document clustering (*Lydia et al., 2020*), text categorization (*Hulth & Megyesi, 2006*), and many more. More specifically, in browsing, searching, and finding similar articles or news reports. These algorithms also find a variety of applications in the field of scientific literature. Several works compare these algorithms for extracting keyphrases from the scientific literature. In *Sarwar & Noor (2021)*, the performance of well-known unsupervised algorithms for extracting keyphrases is compared with scientific literature from the field of computer science. On the other hand, the comparison of supervised and unsupervised keyphrase extraction algorithms is carried out in another study using scientific literature from the Electrical Double Layer Capacitors (EDLC) domain for the extraction of synthesis processes or material properties (*Miah et al., 2021*). However, these algorithms can also be used for keyphrase extraction in other areas of content-based text processing. News article processing is one of the most important tasks in this context. Due to the growing number of online news articles, automatic information extraction from news articles is necessary (*Møller, 2022*). In addition, users can save time reading online news by using automatic keyphrase extraction technologies that can help them identify and remove junk news and quickly find relevant news (*Zhang, 2021*). Extracting important information from news articles is essential for finding similar news items or getting content-based recommendations for news articles (*Sridhar & Sanagavarapu, 2021*). In this case, keyphrase extraction algorithms can play an important role by extracting relevant information from news articles. Since there are several keyphrase extraction algorithms, it is difficult to choose one and apply it to news articles because the writing style varies from content to content. For instance, there are different academic writing styles, such as persuasive, descriptive, narrative, expository, and creative (*Beers & Nagy, 2011*). Therefore, the writing style of news articles differs from that of academic writing because the sentences and paragraphs in academic texts are complex and different from the text of a news article (*Akkaya & Aydin, 2018*). From this perspective, it is reasonable to say that comparing the prominent keyphrase extraction algorithms in news articles is still of great interest.

The most commonly used measure for finding relevant information from news articles is Term Frequency-Inverse Document Frequency (TF-IDF) (*Ding, Zhang & Huang, 2011*; *Lee & Kim, 2008*). TF-IDF is a statistical measure that determines the significance of a keyword by considering its significance in a single document and multiplying it by its significance across all documents in the *corpus*. However, the previous studies show that the other prominent algorithms such as KEA, KP-Miner, TeKET, and Yake perform better than TF-IDF for scientific literature (*Miah et al., 2021*; *Sarwar & Noor, 2021*; *Sarwar et al., 2021*). Therefore, due to the different writing styles of news articles, an extensive experiment is needed to compare the known keyphrase extraction algorithms and select an efficient one. The primary objective of this study is to employ different keyphrase extraction algorithms along with different similarity computation techniques to find similar news articles for a given article. First, we extract keyphrases from news articles using different keyphrase extraction algorithms and calculate their similarity to find similar articles. The employed keyphrase extraction algorithms are KP-Miner, YAKE, TeKET, MR, TR, and KEA. To calculate the similarity between the extracted keyphrases, three prominent similarity calculation techniques are also used, namely Cosine similarity (*Gunawan, Sembiring & Budiman, 2018*), Jaccard similarity (*Niwattanakul et al., 2013*), and Cosine similarity with Word2Vec (*Mikolov et al., 2013*). Cosine similarity with Word2Vec computes the semantic relatedness between the extracted keyphrases. In summary, the significant contributions of this work are:

- A comprehensive experiment is conducted to automatically find similar news articles by using keyphrase extraction algorithms with lexical and semantic similarity approaches.
- A comparative analysis between supervised and unsupervised algorithms is performed to extract high-quality keyphrases from news articles.
- A comparison between lexical and semantic similarity techniques for finding similar news articles is performed.

The remaining part of this article is arranged as follows. The Related Study section briefly discusses keyphrase extraction algorithms and similarity calculation techniques. The Methodology section presents the functionality of the proposed approach in detail. The Experimental Details and Result Discussion section discusses the details of the experimental setting, the result analysis of the experiment, and the discussion of the findings of this study. Finally, the Conclusion section concludes the study with an outlook for the future.

MATERIALS AND METHODS

This study shows the performance comparison of some prominent keyphrase extraction algorithms in terms of calculating the text-similarity as well as semantic relatedness between the extracted keyphrases to find similar news articles. Therefore, this section provides a concise overview of the keyphrase extraction algorithms used as well as the similarity computation techniques.

Keyphrase extraction algorithms

In this study, two types of keyphrase extraction algorithms are used, namely the supervised and unsupervised approaches. The difference between these two approaches is whether the learning process involves a labeled training set or not.

Supervised keyphrase extraction algorithm

The supervised approach (*Turney, 2002*) transforms the keyphrase extraction work into a classification or regression problem (*Wang & Wang, 2019*). It employs the learned model to identify if a candidate phrase in a text is a keyphrase by training it on the labeled training set. The supervised approach needs a large amount of training data to extract good quality keyphrases. In this study, one of the most popular and prominent supervised approaches called KEA (*Witten et al., 1999*) is employed.

KEA: KEA is one of the most well-known supervised keyphrase extraction algorithms developed so far (*Witten et al., 1999*). In KEA, training documents are used to generate a classifier according to the Naive Bayes theorem. In the training phase of the algorithm, a model is created using a labeled dataset in which the words from a set of documents are labeled as keyphrases, and the model is then used to extract keyphrases from the new documents (*Witten et al., 1999*; *Zakrzewska & Matańka, 2006*). KEA analyzes the incoming text for orthographic boundaries such as punctuation marks and line breaks in order to locate suitable phrases. Two features, namely TF-IDF and the first occurrence of the word, are used to evaluate each candidate phrase: TF-IDF and the first occurrence of the term. The prediction model is the first product of the machine learning model. Following that, the keywords are retrieved using this prediction model.

Unsupervised keyphrase extraction algorithm

Since annotated data are not always accessible or easy to obtain, methods for unsupervised keyphrase extraction continue to evolve. Moreover, previous studies have shown that most efforts to manage Big Data use unsupervised algorithms. Therefore, this study examines five state-of-the-art algorithms based on their purported performance, namely KP-Miner, YAKE, TeKET, TopicRank, and MultipartiteRank.

KP-Miner: KP-Miner (*El-Beltagy & Rafea, 2009*) is a very well-known and well-performed statistical-based unsupervised keyphrase extraction algorithm. The keyphrase extraction by KP-Miner is a three-step procedure that includes a selection of candidate keyphrases, weight calculation of candidate keyphrases, and refining the keyphrases. The algorithm KP-Miner follows a ranking procedure that utilizes a modified version of TF-IDF and works with N-Gram. HERE, for N-Gram with the value of $N > 1$, the document frequency is considered to be 1. The weights of multiword candidate keyphrases are likewise increased in proportion to the ratio of single-word candidate keyphrase frequencies to all candidate keyphrase frequencies in KP-Miner.

YAKE: YAKE (*Campos et al., 2020*) is another well-known unsupervised statistical-based keyphrase extraction algorithm that takes advantage of statistical context. YAKE extracts contextual information and word dispersion across the article using unique statistical criteria in addition to the term's position/frequency. YAKE splits the text into different words before preprocessing. Then, for each individual term, a set of five properties is determined: casing, word frequency, word placement, word connectivity to context, and word difference in sentences. The score for each word is then calculated by considering all of these factors. Finally, a three-gram sliding window is utilized to create a continuous succession of one-gram, two-gram, and three-gram candidate keyphrases.

TeKET: TeKET (*Rabby et al., 2020*) is an unsupervised tree-based keyphrase extraction algorithm. TeKET is a domain-independent algorithm that requires no training data and relies on minimum statistical knowledge. TeKET's keyphrase extraction procedure is separated into three phases: candidate keyphrase selection, candidate keyphrase processing, and final keyphrase selection from the candidate keyphrases. TeKET employs the KePhEx (*Rabby et al., 2018*) binary tree, which can extract final keyphrases from the candidate keyphrases. It also employs a novel keyphrase ranking strategy that uses a value called the Cohesiveness Index (CI), which represents the cohesiveness of a word concerning its root in a keyphrase. Therefore, TeKET extracts a large number of keyphrases from the candidate keyphrases.

TopicRank: TopicRank (TR) (*Bougouin, Boudin & Daille, 2013*) is another well-known graph-based unsupervised keyphrase extraction algorithm. To extract candidate phrases, the text is first preprocessed. The candidate phrases are then divided into different topics by hierarchical agglomerative clustering (*Sasirekha & Baby, 2013*). The next step is to create a topic graph, where the edges are weighted according to a metric that takes into account the offset positions of the phrases in the text. The topics are then ranked using TextRank (*Mihalcea & Tarau, 2004*), and a candidate is selected from each of the top N topics.

MultipartiteRank: MultipartiteRank (MR) (*Boudin, 2018*), which is similar to TopicRank, is a well-performing graph-based unsupervised keyphrase extraction algorithm. This algorithm selects possible keyphrases in two steps: first, it converts the entire document into a graph and then assigns a relevancy score to each word. This algorithm is more complex since it includes a phase in which edge weights are adjusted to account for positional information, resulting in a bias toward prospective keyphrases that appear earlier in the text. There are no links between nodes unless they belong to different topics. Thus, a fully directed multipartite graph is created. This algorithm outperforms previous graph-based algorithms by successfully exploiting the strengthening of relationships between topics and candidate keyphrases.

Similarity calculation techniques

The search for similar news articles is one of the primary tasks of this study. In this context, the extracted keyphrases are used for similarity calculation. Similarity can be calculated in several ways. One is to calculate the lexical similarity between the extracted keyphrases (*Maheshwari et al., 2017*). Another is the semantic similarity computation (*Sitikhu et al., 2019*). For lexical similarity computation, the similarity measures Jaccard (*Niwattanakul et al., 2013*) and Cosine (*Gunawan, Sembiring & Budiman, 2018*) are used. For semantic similarity computation, a word embedding approach called Word2Vec (*Mikolov et al., 2013*) is used with the Cosine similarity measure.

Jaccard similarity

The Jaccard similarity measure is well known and considered a lexical similarity measure that calculates the similarity between two keyphrases. Jaccard similarity analyses two sets of keyphrases and calculates the similarity between all pairs of sets by comparing which data are distinct and which are common. Jaccard similarity is uninformed of the true meaning of the word or complete sentence. Jaccard similarity can be calculated by employing Eq. (1).

$$JS(X,Y) = \frac{|X \cap Y|}{|X| + |Y| - |X \cup Y|} \tag{1}$$

Herein, *JS* denotes the Jaccard similarity score. X and Y denote the two sets of keyphrases extracted from the news articles. Here, the value of *JS* varies between 0 and 1 depending on the similarity score between the two news articles.

Cosine similarity

The Cosine similarity measure is a frequently employed similarity measure that is established on Euclidean distance (*Jeong, Yoon & Lee, 2019*). The Cosine similarity measure calculates the distance between two vectors in a multidimensional space by using a dot product to calculate the angle $(cos(theta))$. Although the lengths of the two articles can be drastically different, there is a high probability that they are comparable due to the shorter angle, resulting in a higher similarity score. Cosine similarity can be calculated by employing Eq. (2).

$$CS(X,Y) = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} (X_i)^2}\sqrt{\sum_{i=1}^{n} (Y_i)^2}} \tag{2}$$

Herein, *CS* denotes the Cosine similarity score. X and Y denote the two sets of keyphrases extracted from the news articles and converted into vectors. Here, the value of *CS* varies between 0 and 1 depending on the similarity score between the two news articles.

Semantic similarity using word embedding

The most popular method of calculating the similarity between two texts is semantic similarity measurement (*Bag, Kumar & Tiwari, 2019*), which calculates the similarity of their meaning or calculates the meaning in context. The semantic similarity between news articles is calculated using the idea of vector representation of words (*Jin, Zhang & Liu, 2018*). Word vectors are the mathematical representation of multiple words with comparable values when they frequently occur in a language (*Sitikhu et al., 2019*). It is a trained text representation where words with similar meanings are defined in the same vector space. Word embeddings are formed using a large data *corpus* to train a neural network.

Word2Vec (*Mikolov et al., 2013*), which creates vector representations of words, is one of the most widely used methods for word embedding. The Word2Vec algorithm converts text into word vectors, which may subsequently be used to train any other word to obtain its vector value. Word2Vec captures word associations from a large text *corpus* and stores them in a model that has already been trained. To increase the efficiency of similarity computation, this kind of pre-trained model can find synonyms or semantically related words (*Mikolov et al., 2013*). The Word2Vec model uses the continuous bag-of-words (CBOW) and continuous skip-gram models to learn distributed representations of words with low computational complexity. The CBOW model learns the embedding by predicting the existing word based on its context. The continuous skip-gram model learns given an existing word by predicting the neighboring words. In this study, the continuous skip-gram model is used. Figure 1 depicts the skip-gram training model. The Cosine

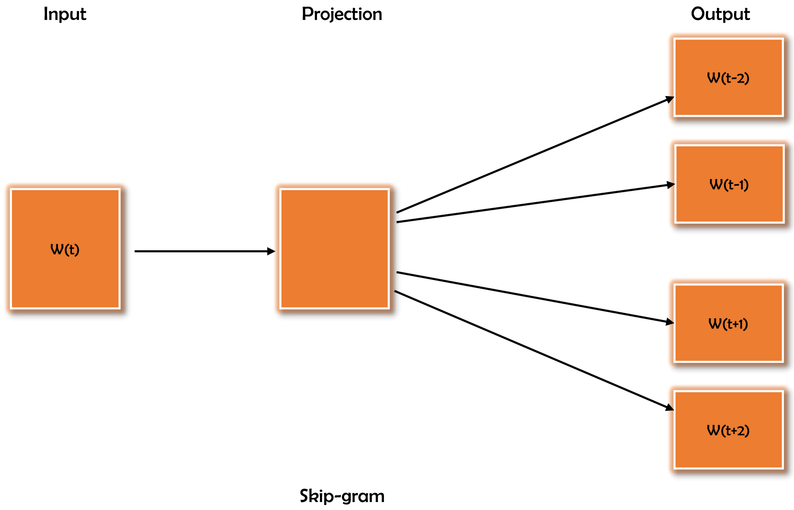

Figure 1 Word2Vec training model skip-gram. This model takes a word as input and tries to predict the similar words based on its context. Full-size DOI: 10.7717/peerj-cs.1024/fig-1

similarity measure is used to calculate the similarity after the word vector values generated by the Word2Vec model (*Jatnika, Bijaksana & Suryani, 2019*).

Methodology

This section provides a comprehensive overview of the methodology for finding similar news articles using keyphrase extraction algorithms and similarity computation techniques. The whole methodology can be split into three stages, namely *i*) data acquisition and pre-processing *ii*) keyphrase extraction, and *iii*) similarity calculation and finding similar articles. A detailed overview of the methodology is shown in Fig. 2.

Data acquisition and pre-processing

This study uses a dataset of news articles collected through the Google News Aggregator service (*Cobos, 2017*). Because coronavirus is a global pandemic, there are many news articles online about coronavirus worldwide that may be relevant to each other. This may help to justify our proposed approach to find more accurate and similar articles for a given article. Therefore, in this study, we select news articles related to coronavirus. To collect the dataset from Google News Aggregator, the authors develop a Python-based News Collector module. This news collector module takes a date range and a topic name as input. Then it collects all relevant news about that topic from all possible newspaper sources within the specified time period. The News Collector module collects the headline, the text of the news article, the publication date, the summary of the article, the URL of the article source, and any media associated with the article. After this information is collected for each news article, it is stored in a dataframe (*The Pandas Development Team, 2021*) and then converted to Microsoft Excel format.

To prepare the dataset for this study, the date range of 1st August 2020 to 30th June 2021 is selected, and "coronavirus" is specified as the topic. Thus, this dataset contains newspaper articles about coronavirus for eleven months. After collecting the data using the

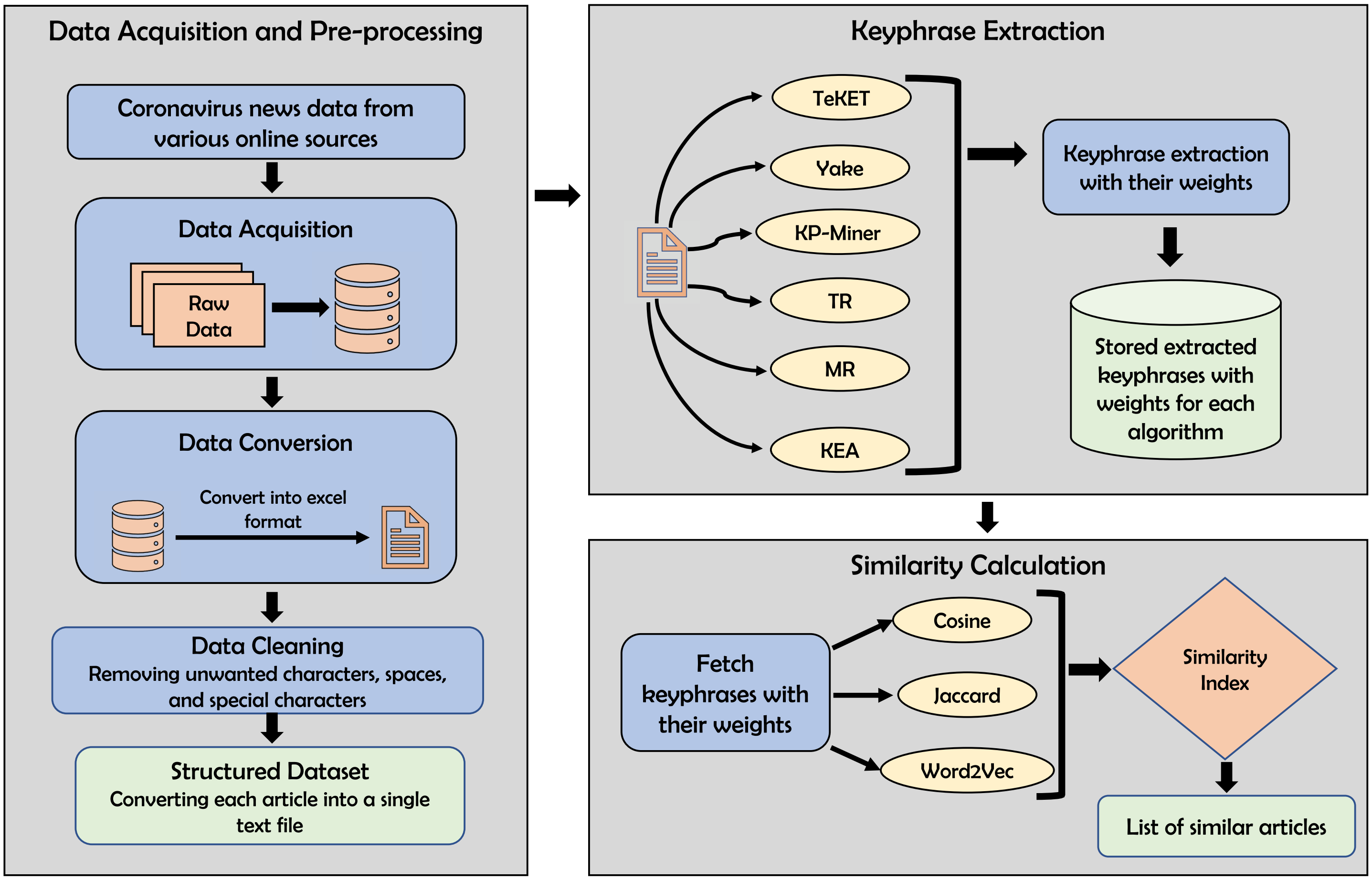
Data Acquisition and Pre-processing
Coronavirus news data from various online sources
Data Acquisition
Raw Data
Data Conversion
Convert into excel format
Data Cleaning
Removing unwanted characters, spaces, and special characters
Structured Dataset
Converting each article into a single text file
Keyphrase Extraction
TeKET
Yake
KP-Miner
TR
MR
KEA
Keyphrase extraction with their weights
Stored extracted keyphrases with weights for each algorithm
Similarity Calculation
Fetch keyphrases with their weights
Cosine
Jaccard
Word2Vec
Similarity Index
List of similar articles

Figure 2 **Functional details of the proposed methodology for finding similar articles.** Full-size DOI: 10.7717/peerj-cs.1024/fig-2

News Collector module, the data is stored in a Microsoft Excel file where each row contains details about a single news article. Each news item is converted to a single text file containing the headline and news text from the Excel file. Before conversion to a single text file, the data is cleaned to remove unwanted characters, spaces, and special characters, and the text data is converted to lowercase.

Keyphrase extraction

After preparing and pre-processing the dataset in the previous step, this step extracts the keyphrases from the collected news articles about coronavirus. A news article is first selected from the dataset to find similar and relevant news articles. The goal is to analyze this article and the other articles in the dataset for similarity. For this purpose, the keyphrases are first extracted from these text documents. The keyphrases are extracted in three steps: *i*) candidate keyphrase selection, *ii*) candidate keyphrase weighting, and *iii*) selecting the final keyphrases from the candidate keyphrases. In this context, several keyphrase extraction algorithms are used. From the supervised approach, KEA is used. From the unsupervised approach, KP-Miner, YAKE, TeKET, MR, and TR are used. At this stage, all the employed keyphrase extraction algorithms return the keyphrases along with their calculated weights.

Similarity calculation and finding similar articles

In this step, the similarity between the targeted main article and other articles is calculated. Since there are different approaches for calculating similarity, both the lexical and semantic similarities are calculated in this study to find more similar and relevant articles. For lexical similarity calculation, Cosine and Jaccard measures are used. The weights of the extracted keyphrases are used to calculate the Cosine and Jaccard similarities between news articles. A word embedding-based approach called Word2Vec is used with Cosine similarity to calculate the semantic similarity between the extracted keyphrases from the news articles. The whole procedure for keyphrase extraction and similarity calculation can be found in Algorithm 1. The most similar and relevant articles can be selected by comparing the calculated similarity scores using different keyphrase extraction algorithms and similarity calculation techniques.

EXPERIMENTAL DETAILS AND RESULT DISCUSSION

Extensive experiments and detailed evaluation are performed to evaluate the proposed approach. The experimental details and experimental results are explained in detail in the Experimental Details and Result Discussion sections, respectively.

Experimental details

This section discusses a comprehensive overview of the experimental setup along with evaluation metrics that are utilized to evaluate the performance of the overall approach for finding similar news articles. The experimental setup and evaluation metrics are presented in Section Experimental Setup and Section Evaluation Metric, respectively.

Experimental setup

The Python programming language is utilized to implement the proposed technique. The version of Python 3.7 is utilized. Stopwords, word_tokenize, and sent_tokenize of Natural Language Toolkit (NLTK) (*Loper & Bird, 2002*) and other related Python packages like math (*Python Software Foundation, 2021a*) and os (*Python Software Foundation, 2021b*) are utilized. The Python Keyphrase Extraction Toolkit (pke) (*Boudin, 2016*) is utilized to implement the statistical-based and graph-based algorithms. TeKET (*Rabby, 2020*) is utilized for the tree-based algorithm. The experiment is performed on a MacBook Pro with a 2.3 GHz quad-core Intel Core i5 processor and 8 GB RAM running macOS Big Sur version 11.6.

Evaluation metric

Since the main objective of this study is to find similar news articles, it is imperative to measure the overall performance of the proposed approach. Therefore, the proposed approach is evaluated using a well-known evaluation metric called Normalized Discounted Cumulative Gain (NDCG) (*Yining et al., 2013*). This evaluation metric is widely used in many areas of article recommendation (*Zhang & Li, 2010*). NDCG is the weighted

Algorithm 1 Algorithm for extracting keyphrases and similarity calculation

```
Input: main article ($M_A$), other articles ($S_A$) s, keyphrase extraction algorithm name ($KP_{algo}$)
Output: similar articles ($A_{sim}$)s
initialize $CS_{List} \leftarrow$ NULL
initialize $JS_{List} \leftarrow$ NULL
initialize $WV_{List} \leftarrow$ NULL
Select $M_A$
extract ($K_{M_A}$)s along with ($W_{K_{M_A}}$)s from $M_A$ using $KP_{algo}$)
/* Here, $K_{M_A}$ are extracted keyphrases from $M_A$ and $W_{K_{M_A}}$ are the weights. */
for $\forall S_A \in (S_A)$s do
    extract ($K_{S_A}$)s with ($W_{K_{S_A}}$)s from $S_A$ using $KP_{algo}$)
    calculate $CS$ using Eq. (2), employing ($W_{K_{M_A}}$)s and ($W_{K_{S_A}}$)s
    make a tuple, $t_{cs}$ using ($articleName$, $CS$)
    append $t_{cs}$ in $CS_{List}$
    calculate $JS$ using Eq. (1), employing ($K_{M_A}$)s and ($K_{S_A}$)s
    make a tuple, $t_{js}$ using ($articleName$, $JS$)
    append $t_{js}$ in $JS_{List}$
    calculate semantic similarity ($SS$) using cosine similarity with Word2Vec, employing ($K_{M_A}$)s and ($K_{S_A}$)s
    make a tuple, $t_{wv}$ using ($articleName$, $SS$)
    append $t_{wv}$ in $WV_{List}$
end
```

average of the top-rated, similarly relevant news articles related to a given article. The value of NDCG can be calculated using Eq. (3).

$$NDCG_r = \frac{DCG_r}{IDCG_r} \tag{3}$$

Herein, $NDCG_r$ denotes the normalized gain acquired at a given rank r for similar news articles. The total discounted cumulative gain at a given rank r for the similar articles found is denoted by DCG_r. Moreover, $IDCG_r$ is the total ideal discounted cumulative gain at a given rank r, which is a DCG measure denoting the top-ranked similar articles (*Järvelin & Kekäläinen, 2002*). The NDCG value generally normalizes the DCG value by dividing it by the IDCG value. The range of the NDCG value is between 0 and 1. The NDCG value of 1 means perfect system performance, and in this case, similar articles found would be the most relevant ones. The DCG/IDCG value can be calculated using Eq. (4).

$$DCG_r/IDCG_r = \sum_{i=1}^{r} \frac{rel_i}{\log_2(i+1)} \tag{4}$$

Herein, rel_i is the relevancy score at position i for the similar news articles concerning a particular article.

To evaluate the extracted keyphrases using different algorithms, a statistical measure named Fleiss' Kappa (*Kılıç, 2015*) is used to determine the inter-annotator agreement. Fleiss' kappa is used to evaluate the dependability of agreement between a specific number of evaluators when giving category ratings to several variables. Equation (5) is used to measure the Fleiss' kappa score.

$$K = \frac{P - \bar{P}_e}{1 - \bar{P}_e} \tag{5}$$

Herein, the factor $1 - \bar{P}_e$ represents the degree of agreement that is possible above chance, while $P - \bar{P}_e$ represents the degree of agreement that is actually achieved. If there is complete agreement among the evaluators, then $K = 1$. If there is no agreement, then $K <= 0$.

RESULT DISCUSSION

One of the foremost concerns of this study is to investigate different keyphrase extraction algorithms in terms of extracting relevant keyphrases from news articles. Therefore, it is imperative to investigate whether the keyphrase extraction algorithms used can extract good quality keyphrases from news articles or not. In this study, different keyphrase extraction algorithms are used. From the supervised category, the feature-based model KEA is used. From the unsupervised category, the graph-based (MR and TR), tree-based (TeKET), and statistical-based (KP-Miner and YAKE) algorithms are used. An example of extracted keyphrases from a particular news article from the dataset can be found in Table 1.

To the best of the author's knowledge, there is no gold standard keyphrase list for different types of news articles, especially for coronavirus news, to analyze the extracted keyphrases for whether they are relevant to the articles or not. Moreover, there are different criteria for writing news articles on a variety of topics. Therefore, creating a gold standard keyphrase list for a variety of topics requires immense manual labor, which is also time consuming. Therefore, the extracted keyphrases for each algorithm are manually evaluated to ensure that they are relevant and summarize the overall concept of the articles. A group of five postgraduate students from the departments of computer science and mechanical engineering evaluated the extracted keyphrases using different algorithms. Among the five students, one is from the department of Mechanical Engineering to avoid bias in the evaluation process. The top-15 extracted keyphrases from the top 10 articles for each algorithm are manually evaluated, and the Fleiss' kappa value is shown in Table 2 for the extracted keyphrases as a measure of Inter Annotator Agreement (IAA).

For visual understanding, a word cloud representation is created after extracting the keyphrases for various news articles. Word clouds are visual representations of words that highlight words that occur more frequently in a text document or that are more prominent due to their rank in a document (*Roe, 2018*). In this study, the word clouds are generated from the extracted keyphrases using different algorithms to investigate the

Table 1 **Example of extracted keyphrases using different algorithms for a particular paper.**

News title	Algorithm name	Extracted keyphrases
It's not just delta–other coronavirus variants worry scientists	KEA	Variants, gamma, vaccinated, antibody, delta, people, cnn, vaccine, gamma variant, variant, according, vaccination, said, going, told, health, fully, coronavirus, states, lindquist, evade, infection, fully vaccinated, told cnn, concerned
	KP-Miner	Variants, gamma, gamma variant, variants worry scientists, delta, delta variant, vaccination, going, said, variant, cnn, health, told, seen, last, state, also, one, lower antibody, tropical medicine, antibody treatments, transmissibility, vaccination rates, resistant
	YAKE	Gamma, gamma variant, scott lindquist, variants, gamma and delta, department of health, moore said, said, cnn that delta, variants worry scientists, alpha and delta, told cnn, spread of variants, variant of concern, delta, transmissibility of gamma, epidemiologist for washington, moore, cnn, coronavirus variants worry, health, moore told cnn, antibody, seen in india, cdc has variant
	MultipartiteRank	Coronavirus variants, gamma, delta variant, variants, lower antibody effectiveness, cnn, washington state, federal health officials, delta, people, antibody treatments, variant, vaccine experts, state, last thing, tropical medicine, cdc, concern, moore, gamma variant, much ability, epidemiologist, resistant, alpha variant
	TopicRank	Coronavirus variants, gamma, delta variant, cnn, delta, people, antibody treatments, washington state, vaccine experts, federal health officials, moore, multiple states, resistant, tropical medicine, transmissible, alpha variant, vaccines, cdc, immunity, single dose, concern, full vaccination, low vaccination rates, particular monoclonal antibodies, lower antibody effectiveness
	TeKET	Variant, gamma, cnn, according, moore

Table 2 **Fleiss' values for the extracted keyphrases scored by the annotators.**

Category	Fleiss' kappa score
Keyphrases relevant to coronavirus	0.96
Keyphrases not relevant to coronavirus	0.94

relevance of the extracted keyphrases with respect to their document. Various factors can be used to create word clouds. For instance, term frequency is often used to create word clouds. However, in this study, we use the weights of keyphrases generated by keyphrase extraction algorithms. Each algorithm generates a list of final ranked keyphrases and their weights, where the weights indicate the relevance of the keyphrases and help in the ranking process. After extracting keyphrases from news articles, word clouds are also generated to see and evaluate the relevance of the extracted keyphrases visually. Figure 3 shows an example of the word clouds created for a news article titled *"It's not just Delta-other coronavirus variants worry scientists, also"*.

The word clouds in Fig. 3 is created from the extracted keyphrases from a news article dealing with coronavirus, more specifically, the variant of coronavirus that may be of concern to scientists. Therefore, the extracted keyphrases should contain keyphrases that relate to the context or reflect the title of the article. If we analyze the word clouds from Fig. 3, we can see that the most common keyphrases extracted from the word clouds are variants, gamma, gamma variant, vaccination, and many more. So we can say that the keyphrase extraction algorithms extract relevant keyphrases concerning the context of the

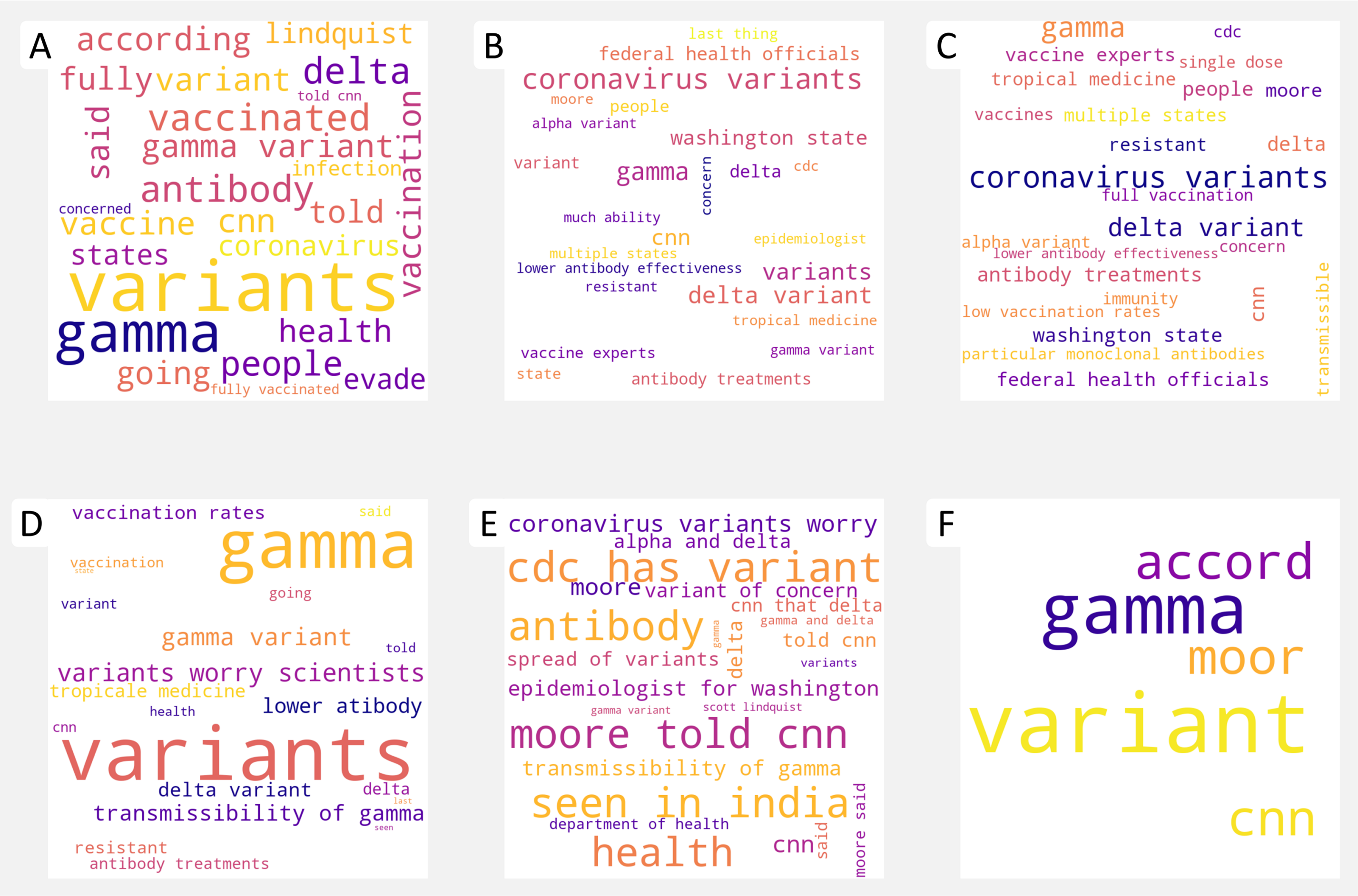

Figure 3 An example of word clouds generation with extracted keyphrases by employing different algorithms, namely KEA (A), MR (B), TR (C), KP-Miner (D), YAKE (E), and TeKET (F) for the comparative analysis. Full-size DOI: 10.7717/peerj-cs.1024/fig-3

articles. However, one notable observation is found from the extensive experiment. Looking at Fig. 3, we can see that all the keyphrase extraction algorithms except TeKET extract a good number of high-quality keyphrases. However, TeKET performs well on scientific literature in terms of extracting high-quality keyphrases (*Sarwar et al., 2021*). TeKET computes a cohesive index (CI) between words to extract the final keyphrases. The CI indicates the degree of cohesiveness between words. However, the scientific literature is much longer than a news article, and the degree of cohesiveness is lower than scientific literature. Therefore, TeKET fails to generate a good number of keyphrases from news articles. Since the extracted keyphrases are used to calculate the similarity index to find similar news articles, failure to extract a good number of keyphrases may result in an inconsistent similarity score. Thus, if the similarity between a few keyphrases is calculated, the chances of getting a high score are much higher. Therefore, TeKET is not considered further in this study when calculating the similarity score for finding similar news articles.

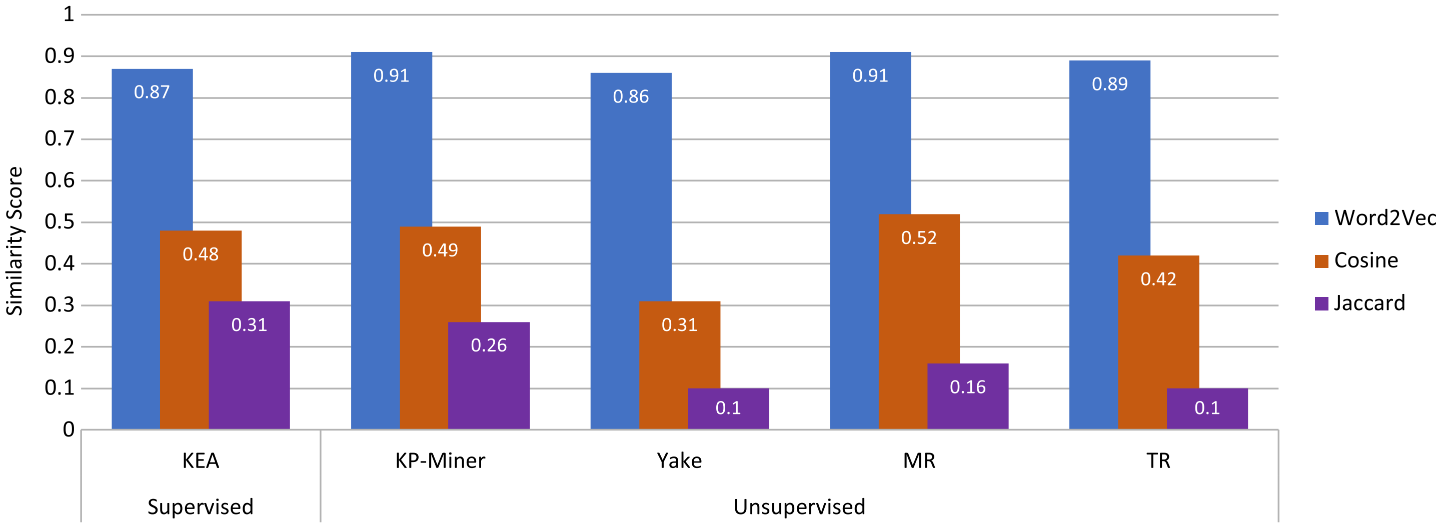

Figure 4 Comparison of the employed keyphrase extraction algorithms along with different similarity calculation techniques.

Full-size DOI: 10.7717/peerj-cs.1024/fig-4

After the keyphrases are extracted, they are used for the similarity calculation. The similarity calculation part is essential in this study because it finds similar messages based on the similarity score. The similarity is calculated based on the extracted keyphrases by using the best performing keyphrase extraction algorithms except for TeKET since TeKET has already been disregarded due to its poor performance. Similarity is computed both lexically and semantically. For the lexical similarity calculation, the cosine and Jaccard measures are used. For semantic similarity computation, on the other hand, the concept of word embedding is used by employing Word2Vec with cosine similarity. In Fig. 4 you can see a comparative analysis of the different techniques used for similarity computation along with the algorithms used for keyword extraction. Figure 4 represents the average similarity scores of the top five most similar news articles for a given news article for which the keyphrase extraction algorithms with different similarity measures were used. The top five articles are acquired for the news article titled "It's not just Delta-other coronavirus variants worry scientists".

From Fig. 4, it can be observed that for the graph-based unsupervised keyphrase extraction algorithms, KP-Miner and MR perform better than the other algorithms by using Word2Vec with the Cosine measure, which is a semantic similarity measure. KP-Miner and MR produce the highest average similarity score of 0.91 using Word2Vec with the Cosine measure. On the other hand, TR produces the second-highest similarity score of 0.89, and the statistical-based approach YAKE produces a similarity score of 0.86 using Word2Vec with the Cosine measure. The supervised approach KEA also performs moderately, producing a similarity score of 0.87 using Word2Vec. On the other hand, for lexical similarity, the Cosine similarity measure performs better in terms of similarity scores than the Jaccard measure utilizing the extracted keyphrases. The average similarity scores between MR and KP-Miner for the Cosine measure have nearly equaled one another. For the Cosine similarity measure, the MR generates an average score of

0.52, while the KP-Miner generates an average score of 0.49. In addition, the MR and KP-Miner generate the Jaccard similarity scores of 0.16 and 0.26, respectively, lower than the Cosine measure. The supervised feature-based algorithm KEA generates an average Cosine and Jaccard similarity scores of 0.48 and 0.31, respectively.

Using Word2Vec with Cosine similarity as a semantic measure result in a higher similarity score than using the Cosine or Jaccard measure. Because, in a document, it keeps the meaning of extracted keyphrases that have similar vector values and lie in the same vector space. Word2Vec also has the advantage of having a smaller embedding vector, unlike other approaches such as Bag of Words or TF-IDF. The Skip-gram model in this experiment also helps to capture the similar vector values of the provided keyphrases to use for the similarity calculation. For this reason, the Cosine similarity measure can produce better similarity scores than the Cosine and Jaccard measures.

The Cosine similarity measure for lexical analysis performs well when the high-dimensional data are in a vector. The magnitude of the keyphrases is the computed weight provided by the keyphrase extraction algorithms used to create vectors for the Cosine similarity measure. Using the weights of keyphrases in the Cosine similarity measure disregards the possibility of using the word count of phrases that frequently occur in the articles but do not necessarily have an impact on being similar articles. This advantage results in the Cosine similarity measure performing better than the Jaccard measure.

On the other hand, the keyphrase extraction algorithms produce lower scores with the Jaccard similarity measure. The main reason behind this is that the number of extracted keyphrases strongly affects the Jaccard similarity measure. If the length of the keyphrase lists is so high with dissimilar keyphrases, then the length of the union list of keyphrases becomes larger while keeping the intersection list unchanged. Therefore, the calculation produces a very low similarity score. The text of a news article may be comparatively long, and the extracted keyphrases may be lexically dissimilar, although they could be semantically related. For this reason, the Jaccard measure produces comparatively lower similarity scores than the Cosine measure.

In part of the experiment, a targeted article from the dataset is compared with the other articles in terms of similarity score to find similar articles. For this purpose, the similarity between the targeted article is computed with the others. First, the keyphrases are extracted from the targeted and compared articles using KEA, KP-Miner, YAKE, MR, and TR. Then, different techniques are used to calculate the similarity score. In this experiment, the top five articles with the highest similarity scores are acknowledged as the most relevant and similar articles. Since the articles with the highest similarity scores are considered as similar articles, the performance of the proposed approach needs to be evaluated. The obtained top-ranked news articles in terms of similarity score should be relevant to the targeted article. Therefore, an expert evaluation is used as a benchmark to evaluate the obtained articles through the mechanism of similarity calculation for these news articles. Expert evaluation can be applied to investigate this kind of automated system that can find or recommend similar articles (*Beel et al., 2013*). Expert evaluation can be an excellent tool for evaluating such an approach, as it can provide insight into the real-time performance of the proposed approach (*Sugiyama & Kan, 2013*). The obtained

Table 3 Different categories with relevancy scores to rank similar news articles.

Category	Score
Not similar	0
Somewhat similar	1
Similar	2
Completely similar	3

Table 4 Performance comparison of the employed different algorithms along with similarity techniques for finding similar articles.

Approach	Algorithm	Similarity technique	NDCG
Supervised	KEA	Cosine Similarity	0.93
		Jaccard Similarity	0.89
		Word2Vec with Cosine Similarity	0.91
	KP-Miner	Cosine Similarity	0.96
		Jaccard Similarity	0.89
		Word2Vec with Cosine Similarity	0.97
	YAKE	Cosine Similarity	0.87
		Jaccard Similarity	0.91
		Word2Vec with Cosine Similarity	0.94
Unsupervised	MR	Cosine Similarity	0.92
		Jaccard Similarity	0.86
		Word2Vec with Cosine Similarity	0.93
	TR	Cosine Similarity	0.82
		Jaccard Similarity	0.79
		Word2Vec with Cosine Similarity	0.91

top five articles identified by different algorithms are manually ranked by the experts by giving them relevancy scores. The ranking made by the experts based on the relevancy scores is considered as the benchmark. The same group of postgraduate students has also participated in the ranking process to give relevancy scores to the top five similar articles identified for a given target article by applying various keyphrase extraction algorithms and similarity calculation techniques. Table 3 shows the relevancy scores, which are divided into four categories.

The performance of the employed approach is then evaluated by comparing the relevancy scores assigned by the experts to the top five articles obtained using different algorithms and similarity techniques. The performance comparison is made by calculating the NDCG values of the different approaches using Eq. (3). The performance comparison of the different algorithms used with similarity techniques is shown in Table 4.

As can be seen in Table 4, the statistical-based algorithm KP-Miner produces the highest NDCG value of 0.97 when using the semantic similarity measure Cosine similarity with Word2Vec, indicating that the top five articles identified using this approach have the highest relevance for a given article. On the other hand, KP-Miner with Cosine similarity measure produces a good NDCG score of 0.96. However, KP-Miner produces a low NDCG score of 0.89 with the Jaccard similarity measure compared to the other two similarity measures. Moreover, all keyphrase extraction algorithms used in this study perform better when combined with the semantic similarity computation approach, namely Cosine similarity with Word2Vec, as they produce high NDCG scores.

YAKE and MR, among the other unsupervised methods, also perform quite similarly when Word2Vec is used with Cosine Similarity. YAKE and MR produce an NDCG value of 0.94 and 0.93, respectively, when using Cosine similarity with Word2Vec. However, the NDCG score differs for YAKE and MR when used with the Cosine and Jaccard similarity measures. MR, which is a graph-based keyphrase extraction algorithm, performs better than the statistical-based algorithm YAKE when combined with the Cosine similarity measure. Using Cosine similarity, YAKE and MR achieve an NDCG value of 0.87 and 0.92, respectively. However, YAKE performs better than MR when using the Jaccard similarity measure. On the other hand, YAKE only performs better than KP-Miner while using the Jaccard similarity measure. On the other hand, YAKE and MR outperform TR in all respects. Yake and MR achieve better NDCG scores than the TR on all similarity measures.

The only supervised keyphrase extraction algorithm KEA performs better with the lexical similarity measure called Cosine similarity. It comes in second place with an NDCG value of 0.93 for the Cosine measure. KEA performs relatively well when compared to the other keyphrase extraction algorithms using two lexical similarity computation techniques, Cosine, and Jaccard. The overall performance of the proposed approach using various algorithms with different similarity calculation approaches is illustrated in Fig. 5. For the semantic similarity approach, it can be observed that the KP-Miner keyphrase extraction algorithm with Cosine-Word2Vec similarity measure outperforms the other approaches for finding similar news articles for a given article. For the lexical approach, KP-Miner with Cosine similarity measure again outperforms other combinations of approaches for finding similar news articles for a given article.

Since the top-performing Keyphrase extraction algorithm is KP-Miner, the top five obtained similar articles by KP-Miner with Cosine-Word2Vec, Cosine, and Jaccard similarity are depicted in Tables 5–7 respectively. For Tables 5–7, the first column denotes the main targeted article for which similar articles will be found. The second column denotes the top five similar articles found for that targeted article. The third column denotes the calculated similarity score with different similarity measures for the top five similar news articles.

For better understanding, Fig. 6 shows the comparison of the similarity scores of the top five articles obtained by KP-Miner along with different similarity measures. From the figure, it can also be seen that Word2Vec produces higher similarity scores than the other two techniques. Therefore, it can be concluded from this study that KP-Miner performs better with Word2Vec among the other keyphrase extraction algorithms and similarity techniques used.

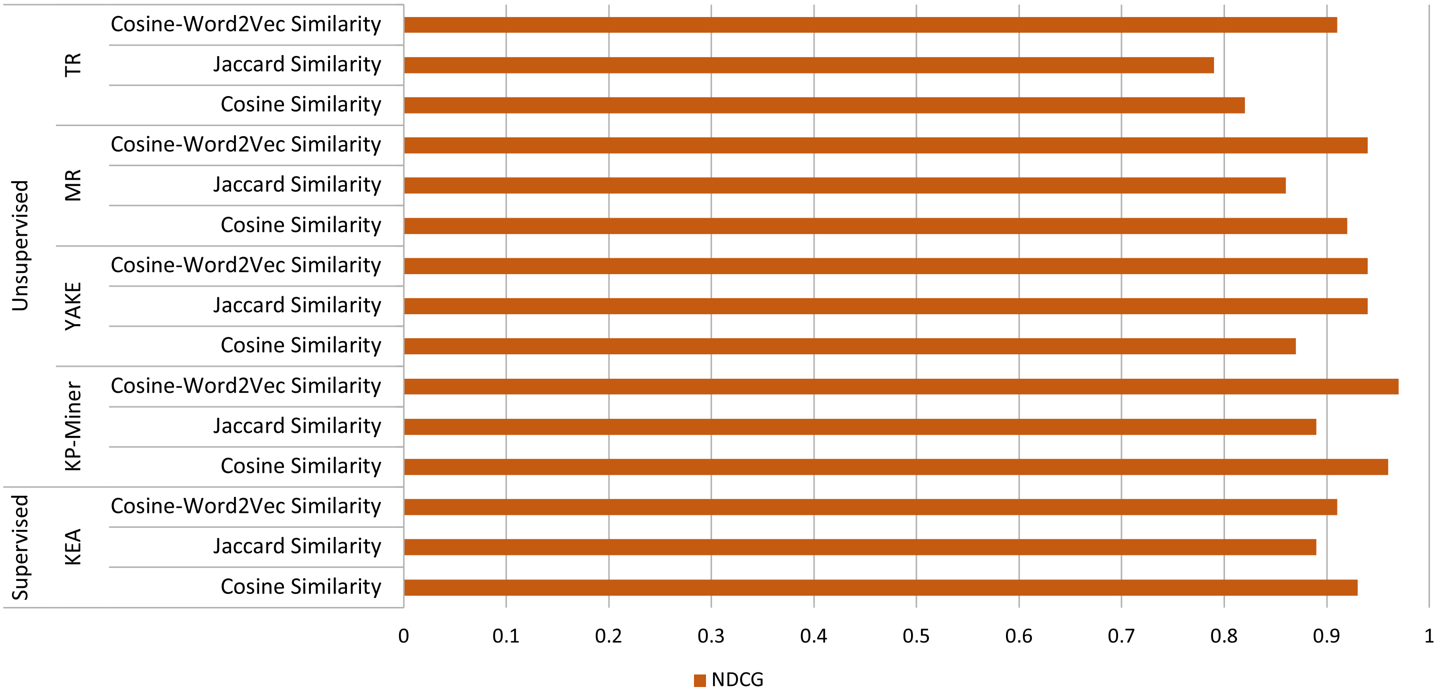

Figure 5 **Performance of the proposed approach (employed keyphrase extraction algorithms along with different similarity calculation techniques) for finding similar news articles.** Full-size DOI: 10.7717/peerj-cs.1024/fig-5

Table 5 **Obtained top five similar news articles for a particular article employing KP-Miner with Cosine-Word2Vec similarity measure.**

Main article	Similar article	Cosine with word2vec
It's not just Delta–other coronavirus variants worry scientists, also	Delta Plus What we know about the coronavirus variant	0.883
	Here's what we know about the Delta variant of coronavirus	0.874
	Explainer: what is the Delta variant of coronavirus with K417N mutation?	0.871
	Fact check What do we know about the coronavirus delta variant?	0.859
	Why No One Is Sure If Delta Is Deadlier	0.813

Table 6 **Obtained top five similar news articles for a particular article employing KP-Miner with Cosine similarity measure.**

Main article	Similar article	Cosine with Word2Vec
It's not just delta–other coronavirus variants worry scientists, also	Coronavirus new variant–genomics researcher answers key questions	0.553
	Coronavirus lambda variant spreads across Latin America	0.553
	Fauci Warns Dangerous Delta Variant Is The Greatest Threat To U.S. COVID efforts	0.473
	Here's what we know about the Delta variant of coronavirus	0.455
	Fact check What do we know about the coronavirus delta variant?	0.448

Table 7 **Obtained top five similar news articles for a particular article employing KP-Miner with Jaccard similarity measure.**

Main article	Similar article	Jaccard similarity
It's not just Delta–other coronavirus variants worry scientists, also	Here's what we know about the Delta variant of coronavirus	0.316
	Explainer: What is the Delta variant of coronavirus with K417N mutation	0.263
	Delta coronavirus variant scientists brace for impact	0.261
	Why No One Is Sure If Delta Is Deadlier?	0.25
	Fauci Warns Dangerous Delta Variant Is The Greatest Threat To U.S. COVID Efforts	0.222

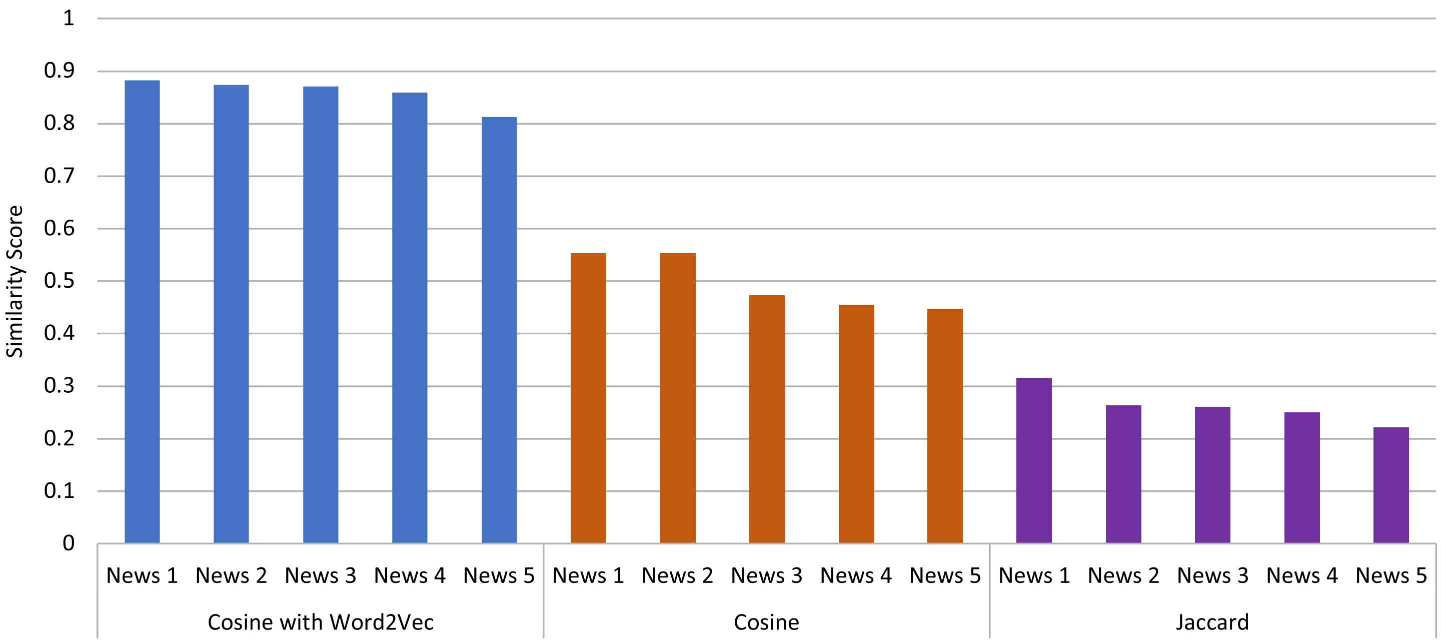

Figure 6 **Comparison of different similarity measures for top five news articles obtained by the KP-Miner algorithm.**

Full-size DOI: 10.7717/peerj-cs.1024/fig-6

CONCLUSIONS AND FUTURE WORK

Keyphrases in a document are considered key concepts and reflect prior knowledge that can be used for a variety of purposes. They can provide a concise summary of the text that can be used for both human and machine-readable activities, such as facet search, text categorization, text clustering, query generation, recommendations, and more.

In this study, we investigate the current state of knowledge on various supervised and unsupervised algorithms for extracting keyphrases for news articles. The study also compares the approach of computing lexical and semantic similarity based on the extracted keyphrases by different keyphrase extraction algorithms to find similar news articles. For the experiment, a dataset on coronavirus is prepared using the Google News Aggregator service. First, the keyphrases along with their weights are extracted using the different keyphrase extraction algorithms. Then, the similarities between the targeted news article and the other news articles are calculated using the lexical and semantic similarity approach. The experiment shows that the unsupervised algorithm KP-Miner with the semantic similarity calculation technique Word2Vec outperforms the other combinations of keyphrase extraction algorithms and similarity calculation techniques. KP-Miner with Cosine-Word2Vec can find the most similar news articles with an NDCG value of 0.97. KP-Miner also performs well with the Cosine similarity measure and achieves an NDCG value of 0.96. Moreover, the supervised algorithm KEA performs moderately with the Cosine similarity measure and achieves an NDCG value of 0.93. On the other hand, YAKE and MR perform moderately with Cosine-Word2Vec and achieve NDCG values of 0.94 and 0.93, respectively.

As the keyphrases extracted with different algorithms are manually evaluated by the IAA procedure, an extensive evaluation will be performed in the future, following an automated systematic evaluation process. Moreover, the acquired similar news articles are also manually ranked by experts to evaluate the performance of the proposed approach. Thus, the experiment is conducted with respect to a specific topic. In the future, this study can be taken further by implementing different topics of news articles where the extracted keyphrases can be classified into different topics. In this way, similar news articles can be recommended to the users depending on their interest in the different topics.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work is supported by the University Malaysia Pahang (UMP) Flagship Grant RDU182201-3. The funding is supported by the university Flagship project. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
University Malaysia Pahang (UMP) Flagship: RDU182201-3.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Talha Bin Sarwar conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Noorhuzaimi Mohd Noor conceived and designed the experiments, authored or reviewed drafts of the article, and approved the final draft.
- M. Saef Ullah Miah performed the experiments, analyzed the data, performed the computation work, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The raw data and code are available in the Supplemental Files.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.1024#supplemental-information.

REFERENCES

Akkaya A, Aydin G. 2018. Academics' views on the characteristics of academic writing. *Educational Policy Analysis and Strategic Research* **13(2)**:128–160 DOI 10.29329/epasr.2018.143.7.

Azad HK, Deepak A. 2019. Query expansion techniques for information retrieval: a survey. *Information Processing & Management* **56(5)**:1698–1735 DOI 10.1016/j.ipm.2019.05.009.

Babar S, Patil PD. 2015. Improving performance of text summarization. *Procedia Computer Science* **46**:354–363 DOI 10.1016/j.procs.2015.02.031.

Bag S, Kumar SK, Tiwari MK. 2019. An efficient recommendation generation using relevant Jaccard similarity. *Information Sciences* **483(6)**:53–64 DOI 10.1016/j.ins.2019.01.023.

Beel J, Langer S, Genzmehr M, Gipp B, Breitinger C, Nürnberger A. 2013. Research paper recommender system evaluation: a quantitative literature survey. In: *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*. 15–22.

Beers SF, Nagy WE. 2011. Writing development in four genres from grades three to seven: syntactic complexity and genre differentiation. *Reading and Writing* **24(2)**:183–202 DOI 10.1007/s11145-010-9264-9.

Boudin F. 2016. Pke: an open source Python-based keyphrase extraction toolkit. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*. 69–73.

Boudin F. 2018. Unsupervised keyphrase extraction with multipartite graphs. *ArXiv preprint* DOI 10.48550/arXiv.1803.08721.

Bougouin A, Boudin F, Daille B. 2013. Topicrank: graph-based topic ranking for keyphrase extraction. In: *International joint conference on natural language processing (IJCNLP)*. 543–551.

Campos R, Mangaravite V, Pasquali A, Jorge A, Nunes C, Jatowt A. 2020. Yake! Keyword extraction from single documents using multiple local features. *Information Sciences* **509(2)**:257–289 DOI 10.1016/j.ins.2019.09.013.

Cobos TL. 2017. New scenarios in news distribution: the impact of news aggregators like google news in the media outlets on the web. In: Tosoni S, Carpentier N, Murru MF, Kilborn R, Kramp L, Risto K, McNicholas A, eds. *Present Scenarios of Media Production and Engagement*. Bremen: Edition Lumière, 95.

Ding Z, Zhang Q, Huang X-J. 2011. Keyphrase extraction from online news using binary integer programming. In: *Proceedings of 5th International Joint Conference on Natural Language Processing*. 165–173.

El-Beltagy SR, Rafea A. 2009. KP-Miner: a keyphrase extraction system for English and Arabic documents. *Information Systems* **34(1)**:132–144 DOI 10.1016/j.is.2008.05.002.

Gunawan D, Sembiring C, Budiman MA. 2018. The implementation of cosine similarity to calculate text relevance between two documents. *Journal of Physics: Conference Series* **978(1)**:12120 DOI 10.1088/1742-6596/978/1/012120.

Hasan KS, Ng V. 2014. Automatic keyphrase extraction: a survey of the state of the art. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1262–1273.

Hulth A, Megyesi B. 2006. A study on automatically extracted keywords in text categorization. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. 537–544.

Jatnika D, Bijaksana MA, Suryani AA. 2019. Word2vec model analysis for semantic similarities in english words. *Procedia Computer Science* **157(4)**:160–167 DOI 10.1016/j.procs.2019.08.153.

Jeong B, Yoon J, Lee J-M. 2019. Social media mining for product planning: a product opportunity mining approach based on topic modeling and sentiment analysis. *International Journal of Information Management* **48**:280–290 DOI 10.1016/j.ijinfomgt.2017.09.009.

Jin X, Zhang S, Liu J. 2018. Word semantic similarity calculation based on word2vec. In: *2018 International Conference on Control, Automation and Information Sciences (ICCAIS)*. Piscataway: IEEE, 12–16.

Järvelin K, Kekäläinen J. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)* **20(4)**:422–446 DOI 10.1145/582415.582418.

Kılıç S. 2015. Kappa testi. *Journal of Mood Disorders* **5(3)**:142 DOI 10.5455/jmood.20150920115439.

Lee S, Kim H-j. 2008. News keyword extraction for topic tracking. In: *2008 Fourth International Conference on Networked Computing and Advanced Information Management*. Piscataway: IEEE, 554–559.

Loper E, Bird S. 2002. Nltk: the natural language toolkit. *ArXiv preprint* DOI 10.48550/arXiv.cs/0205028.

Lydia EL, Kumar PK, Shankar K, Lakshmanaprabu S, Vidhyavathi R, Maseleno A. 2020. Charismatic document clustering through novel k-means non-negative matrix factorization (knmf) algorithm using key phrase extraction. *International Journal of Parallel Programming* **48(3)**:496–514 DOI 10.1007/s10766-018-0591-9.

Maheshwari G, Trivedi P, Sahijwani H, Jha K, Dasgupta S, Lehmann J. 2017. Simdoc: topic sequence alignment based document similarity framework. In: *Proceedings of the Knowledge Capture Conference*. 1–8.

Miah MSU, Sulaiman J, Sarwar TB, Naseer A, Ashraf F, Zamli KZ, Jose R. 2022. Sentence boundary extraction from scientific literature of electric double layer capacitor domain: tools and techniques. *Applied Sciences* **12(3)**:1352 DOI 10.3390/app12031352.

Miah M, Sulaiman J, Sarwar TB, Zamli KZ, Jose R. 2021. Study of keyword extraction techniques for electric double-layer capacitor domain using text similarity indexes: an experimental analysis. *Complexity* **2021(4)**:1–12 DOI 10.1155/2021/8192320.

Mihalcea R, Tarau P. 2004. Textrank: bringing order into text. In: *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.

Mikolov T, Chen K, Corrado G, Dean J. 2013. Efficient estimation of word representations in vector space. *ArXiv preprint* DOI 10.48550/arXiv.1301.3781.

Møller LA. 2022. Recommended for you: how newspapers normalise algorithmic news recommendation to fit their gatekeeping role. *Journalism Studies* **23(7)**:1–18 DOI 10.1080/1461670X.2022.2034522.

Niwattanakul S, Singthongchai J, Naenudorn E, Wanapu S. 2013. Using of jaccard coefficient for keywords similarity. In: *Proceedings of the international multiconference of engineers and computer scientists*. **1**:380–384.

Python Software Foundation. 2021a. math—mathematical functions—Python 3.9.1rc1 documentation. *Available at https://docs.python.org/3/library/math.html*.

Python Software Foundation. 2021b. os—miscellaneous operating system interfaces —Python 3.9.1rc1 documentation. *Available at https://docs.python.org/3/library/os.html*.

Rabby G. 2020. TeKET-Automatic Keyphrase Extraction. GitHub. *Available at https://github.com/TalhaSarwar40/TeKET.git*.

Rabby G, Azad S, Mahmud M, Zamli KZ, Rahman MM. 2018. A flexible keyphrase extraction technique for academic literature. *Procedia Computer Science* **135(3)**:553–563 DOI 10.1016/j.procs.2018.08.208.

Rabby G, Azad S, Mahmud M, Zamli KZ, Rahman MM. 2020. Teket: a tree-based unsupervised keyphrase extraction technique. *Cognitive Computation* **12(4)**:811–833 DOI 10.1007/s12559-019-09706-3.

Roe A. 2018. Generating word clouds. *The School Librarian* **66(1)**:19.

Sarwar TB, Noor NM. 2021. An experimental comparison of unsupervised keyphrase extraction techniques for extracting significant information from scientific research articles. In: *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*. Piscataway: IEEE, 130–135.

Sarwar TB, Noor NM, Saef Ullah Miah M, Rashid M, Farid FA, Husen MN. 2021. Recommending research articles: a multi-level chronological learning-based approach using unsupervised keyphrase extraction and lexical similarity calculation. *IEEE Access* **9**:160797–160811 DOI 10.1109/ACCESS.2021.3131470.

Sasirekha K, Baby P. 2013. Agglomerative hierarchical clustering algorithm-a review. *International Journal of Scientific and Research Publications* **83**:83.

Sitikhu P, Pahi K, Thapa P, Shakya S. 2019. A comparison of semantic similarity methods for maximum human interpretability. In: *2019 Artificial Intelligence for Transforming Business and Society (AITB)*. Piscataway: IEEE, 1–4.

Sridhar S, Sanagavarapu S. 2021. Content based news recommendation engine using hybrid bilstm-ann feature modelling. In: *2021 Joint 10th International Conference on Informatics, Electronics Vision (ICIEV) and 2021 5th International Conference on Imaging, Vision Pattern Recognition (icIVPR)*. 1–8.

Sugiyama K, Kan M-Y. 2013. Exploiting potential citation papers in scholarly paper recommendation. In: *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*. New York: ACM, 153–162.

The Pandas Development Team. 2021. Pandas. DataFrame—pandas 1.3.4 documentation. *Available at https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html*.

Turney PD. 2002. Learning to extract keyphrases from text. *ArXiv preprint* DOI 10.48550/arXiv.cs/0212013.

Wang R, Wang G. 2019. Web text categorization based on statistical merging algorithm in big data environment. *International Journal of Ambient Computing and Intelligence (IJACI)* **10(3)**:17–32 DOI 10.4018/IJACI.

Welleck S, Brantley K, Iii HD, Cho K. 2019. Non-monotonic sequential text generation. In: *International Conference on Machine Learning*. 6716–6726.

Witten IH, Paynter GW, Frank E, Gutwin C, Nevill-Manning CG. 1999. Kea: practical automatic keyphrase extraction. In: *Proceedings of the fourth ACM conference on Digital, libraries*. New York: ACM, 254–255.

Yining W, Liwei W, Yuanzhi L, Di H, Wei C, Tie-Yan L. 2013. A theoretical analysis of ndcg ranking measures. In: *JMLR: Workshop and Conference Proceedings*. **2013**:1–30.

Zakrzewska D, Matańska K. 2006. Automatic keyphrase extraction. *Annales Universitatis Mariae Curie-Sklodowska, sectio AI-Informatica* **5(1)**:101–111 DOI 10.17951/ai.2006.5.1.101-111.

Zha H. 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM, 113–120.

Zhang K. 2021. Web news data extraction technology based on text keywords. *Complexity* **2021(1–2)**:1–11 DOI 10.1155/2021/5529447.

Zhang Z, Li L. 2010. A research paper recommender system based on spreading activation model. In: *The 2nd International Conference on Information Science and Engineering*. Piscataway: IEEE, 928–931.