*mathematics*

MDPI

*Article*

# Keyword Pool Generation for Web Text Collecting: A Framework Integrating Sample and Semantic Information

**Xiaolong Wu [1,2,3], Chong Feng [3,4], Qiyuan Li [1,2] and Jianping Zhu [2,4,5,*]**

[1] School of Medicine, Xiamen University, Xiamen 361105, China; wuxiaolong@stu.xmu.edu.cn (X.W.); qiyuanlii@xmu.edu.cn (Q.L.)
[2] National Institute for Data Science in Health and Medicine, Xiamen University, Xiamen 361105, China
[3] Data Mining Research Center, Xiamen University, Xiamen 361005, China; 2022000119@xmut.edu.cn
[4] School of Mathematics and Statistics, Xiamen University of Technology, Xiamen 361105, China
[5] School of Management, Xiamen University, Xiamen 361005, China
* Correspondence: jpzhuxmu@163.com or xmjpzhu@xmu.edu.cn; Tel.: +86-0592-2182376

**Abstract:** Keyword pools are used as search queries to collect web texts, largely determining the size and coverage of the samples and provide a data base for subsequent text mining. However, how to generate a refined keyword pool with high similarity and some expandability is a challenge. Currently, keyword pools for search queries aimed at collecting web texts either lack an objective generation method and evaluation system, or have a low utilization rate of sample semantic information. Therefore, this paper proposed a keyword generation framework that integrates sample and semantic information to construct a complete and objective keyword pool generation and evaluation system. The framework includes a data phase and a modeling phase, and its core is in the modeling phase, where both feature ranking and model performance are considered. A regression model about a topic vector and word vectors is constructed for the first time based on word embedding, and keyword pools are generated from the perspective of model performance. In addition, two keyword generation methods, Recursive Feature Introduction (RFI) and Recursive Feature Introduction and Elimination (RFIE), are also proposed in this paper. Different feature ranking algorithms, keyword generation methods and regression models are compared in the experiments. The results show that: (1) When using RFI to generate keywords, the regression model using ranked features has better prediction performance than the baseline model, and the number of generated keywords is refiner, and the prediction performance of the regression model using tree-based ranked features is significantly better than that of the one using SHAP-based ranked features. (2) The prediction performance of the regression model using RFI with tree-based ranked features is significantly better than that using Recursive Feature Elimination (RFE) with tree-based one. (3) All four regression models using RFI/RFE with SHAP- based/tree-based ranked features have significantly higher average similarity scores and cumulative advantages than the baseline model (the model using RFI with unranked features). (4) Light Gradient Boosting Machine (LGBM) using RFI with SHAP-based ranked features has significantly better prediction performance, higher average similarity scores, and cumulative advantages. In conclusion, our framework can generate a keyword pool that is more similar to the topic, and more refined and expandable, which provides certain research ideas for expanding the research sample size while ensuring the coverage of topics in web text collecting.

**Keywords:** keyword pool generation; web text collecting; search query; word embedding; feature ranking; feature selection

**MSC:** 68T09



**Citation:** Wu, X.; Feng, C.; Li, Q.; Zhu, J. Keyword Pool Generation for Web Text Collecting: A Framework Integrating Sample and Semantic Information. *Mathematics* **2024**, *12*, 405. https://doi.org/10.3390/math12030405

Academic Editor: Faheim Sufi

Received: 12 December 2023
Revised: 23 January 2024
Accepted: 25 January 2024
Published: 26 January 2024



## 1. Introduction

With the rapid update and iteration of Internet of Things technology, mobile communication technology and terminal devices, the Internet has become a carrier of massive

information transmission. This information not only contains some structured data, but also contains a large and valuable amount of unstructured data, especially text data [1]. Web texts are often presented as social media posts and comments, news, forum posts, e-commerce reviews, literature databases, etc. Currently, web text mining is showing a hot trend in various fields. However, many related studies have not paid enough attention to web text collecting. For example, in the COVID-19 study, one study searched for relevant tweets using only one keyword "coronavirus" [2] while another study used a modified Delphi method to identify 23 search keywords, including synonyms and special words such as "SARS-CoV-2" and "lockdown" [3]. Obviously, the former collected more biased samples, which would directly affect the size and coverage of the samples, and thus indirectly affect the robustness and reliability of the subsequent text mining results. Therefore, the number and quality of keywords used as search queries have a significant impact on the collection of web text samples. Since web text usually originate from some reliable websites (but it does not necessarily mean that the sample is real, such as fake news and fake comments [4], which is another area that needs to be studied), the pool of keywords that are used as search queries determines the size and topic coverage of the sample to a large extent. Failure to pay attention to the completeness and reasonableness of the sample space can lead to biased sample distributions and thus biased results. This is a very important issue that is often overlooked by some researchers, so this study focuses on generating a search keyword pool for web text collecting.

Collecting large amounts of relevant topic-specific web text based on search engine usually requires first generating a keyword pool based on the research topic (or the seed keyword) [5,6]. With this keyword pool as search queries, the data Application Programming Interface (API) provided by the target website is call or a web page crawler is performed on the target website to collect data [7]. After text preprocessing, the target text database is constructed. Thus, the pool of keywords largely defines the sample space for a study. This study delved into how to generate a highly similar to the topic and somewhat expansive keyword pool for web text collecting.

The main difference between this study and previous studies is that we proposed a framework integrating sample and semantic information that contains the data phase and the model phase, with the core being the modeling phase, i.e., feature ranking based on word embedding and constructing regression models about the topic vector and word vectors. Specifically, the study ranks the words by a ranking algorithm based on word embedding, trains a base regression model about the topic vector and the word vectors using the word order (feature ranking), and then finds the corresponding keyword pool based on the minimum average loss of the regression model after multiple rounds of training.

The main contributions of this study are as follows:

(1) A framework integrating sample and semantic information for keyword pool generation was proposed, which includes both a data phase and a model phase.
(2) Two kinds of keyword generation methods, Recursive Feature Introduction (RFI) and the Recursive Feature Introduction and Elimination (RFIE), were proposed.
(3) This paper used the feature ranking algorithm based on word embedding to construct regression models about the topic vector and word vectors for the first time, and generated keyword pools from the perspective of model performance (i.e., model average loss).
(4) The experimental results show that, comparing different feature ranking algorithms, keyword generation methods, and regression models, Light Gradient Boosting Machine (LGBM) using RFI methods with SHAP-based ranked features (SHAP-based + RFI) not only performs best in terms of prediction performance, but also its generated keyword pools perform best in terms of average similarity scores and cumulative similarity scores.

The rest of this paper is organized as follows. After providing an in-depth analysis of previous studies in related works in Section 2, the paper proposed a framework in Section 3.

Experiment was given in Section 4, while in the following Section 5 result analysis was provided. And we made conclusions and a plan for future work in Section 6.

## 2. Related Works

As it is mentioned in introduction, some researchers often neglect the completeness of the sample space in the process of collecting web text data, i.e., the keyword pool generated for search is obviously insufficient to cover the topic. However, there are studies showing that researchers in the advertising field have made some progress in this area, such as Sponsored Search Advertising (SSA). Due to the specificity of SSA, researchers aim to generate a wider range of keyword pool to effectively attract potential consumers [8] and lower the advertising cost [9]. However, in web text mining, the generation of keyword pools is only a less noticeable stage in web text collecting, and researchers are more concerned with mining the text obtained from search queries based on keyword pools. In fact, the sample text obtained from keyword-based search queries increases exponentially with each additional keyword, while considering the sample coverage, so text mining related research is even more in need of refined (fewer and better) keyword pools. In this section, the literature related to the keyword pool generation were reviewed.

The first type of keyword pool is directly using the topic. For example, to build an intelligent COVID-19 early warning system, Zhang et al. [2] used Twitter data collected from a subset of a public dataset and tweets searched with the keyword "coronavirus" for machine learning. Cronin et al. [10] collected Tweets containing the hashtag #Shutdown-Stories during the U.S. government shutdown from 2018 to 2019, and explored networked care and the temporal aspects of social media activism by text mining with manual content analysis. Michalko et al. [11] collected text documents from the Newton media by using keywords "dementia" and "Alzheimer's disease", and adopted social network analysis to explore dementia representations in the Slovak media.

The second kind of keyword pool is generated through reviewing literatures related to the topic, or the decision of an expert group (usually by experts scoring or voting). Generating a keyword pool in this way is also by far the most common search queries for web text mining. For example, Zhao et al. [12] identified a pool of 30 keywords through literature review, and used them to collect posts related to illicit drugs as experimental data from several social media platforms. Hung et al. [3] generated 23 potential keywords based on a modified Delphi method, then collected relevant tweets on Twitter and used social networks to analyze changing public sentiment during COVID-19. After the literature review, Wu et al. [13] determined 10 keywords with high social concern about organ donation and transplantation by modified Delphi method, and then used them as search queries to collect relevant news, and later mined the important current status and problems related to the topic.

The third type of keyword pool is generated by analyzing and discussing sample texts. For example, Chen et al. [14] first collected a small sample of 311 rumor messages related to COVID-19 through the Sina Weibo (Chinese Twitter) Official Account to Refute Rumors, then performed word frequency analysis and discussion to identify a pool of five keywords, and finally used text mining to assess the prevalence of rumors and official responses during the COVID-19 outbreak in China. In addition to setting the keywords "privacy" and "confidentiality" as the search queries, Bhatt et al. [15] used a topic model created from COVID-19-related web pages that provide health updates to specifically match keywords about health. The authors collected tweets based on this two-part keyword pool and then used document clustering to analyze privacy concerns in the context of big data, especially during a pandemic like COVID-19. Barchiesi et al. [16] extracted keyword pools by analyzing the core value statements of some well-known companies, and then searched and collected relevant tweets for text mining, social network and assessing stakeholders' attitudes towards the company's core values.

The fourth type of keyword pool is generated by statistical information-based methods and conceptual hierarchy-based methods, and these methods are commonly used in the

field of advertising [8]. A novel approach proposed by Joshi et al. [9], TermsNet, leverages search engines based on word co-occurrence to determine relevance between terms and translate their semantic relationships into a directed graph. By observing a term's neighbors in the graph, the authors generated the common and the nonobvious keywords that were relevant to a term. Chen et al. [17] proposed a novel method for obtaining a keyword pool based on concept hierarchies. The authors first expanded the connotation of the seed keyword through concept matching and their hierarchy, and then leveraged the statistical co-occurrence of the concept information to recommend new keywords. Zhang et al. [18] leveraged content-based PageRank on the Wikipedia graph to rank relevant entities and added an advertising-based factor to the model to recommend advertising keywords for short-text web pages. Zhou et al. [19] generated a target keyword pool based on a seed keyword using a generative neural network model. This keyword pool is not only relevant to the input, but the domain categories of the generated keywords are also consistent with the expectations. Nie et al. [8] first constructed an articles graph based on the link structure of Wikipedia in an iterative manner, and identified the connections between articles by a modified Spreading Activation algorithm, and then extracted keywords from Wikipedia articles based on a novel Bayesian keyword weighting method.

By reviewing and summarizing the first three types of keyword pool generation approaches, this paper found that in most of the previous text mining related studies, the keyword pools used as search queries were generally generated empirically, and a few researchers generated keyword pools based on sample discussions or analysis. However, these keyword pools generation approaches generally lack an objective evaluation process and rarely utilize the semantic information of the sample texts. For the fourth type of keyword pool generation approach, previous studies mainly focused on the field of advertising, which aim at generating keyword pools, and the more the better, which is the biggest difference with the field of text mining. This kind of research focuses on the semantic relationship in the process of keyword generation and the keyword pool generation method, and there is a complete method system, but it lacks the in-depth analysis of the sample text. Therefore, for the field of text mining, this paper integrates the third and fourth types of keyword pool generation approaches, i.e., to build a complete and objective keyword pool generation and evaluation system with the keyword generation method and semantic relationship as the core, and focusing on the semantic information of the sample text.

## 3. Methods

### *3.1. Framework Overview*

The main objective of this framework is to propose a refined keyword pool generation system with high similarity to the topic and some extensibility to provide a solid foundation for web text collecting and mining. The framework integrates sample and semantic information and consists of two main phases, namely data phase and model phase. The overview of the framework architecture is shown in Figure 1. In the data phase, an investigation method for web text collecting based on the target topic and websites is performed firstly. Then comes the preprocessing of the text. In the model stage, a vectorized representation of the text with a word embedding model is performed firstly. Next, based on the topic vector, all the word vectors are ranked by a ranking algorithm. Then, word vectors are introduced into or eliminated from a base regression model about the topic vector and the word vectors based on word order, and the average loss of the model after words are introduced into or eliminated from the model is computed after each round of training. And the corresponding keyword pool based on the minimum average loss of the models is obtained. Finally, the effectiveness of the keyword pool needs to be evaluated.

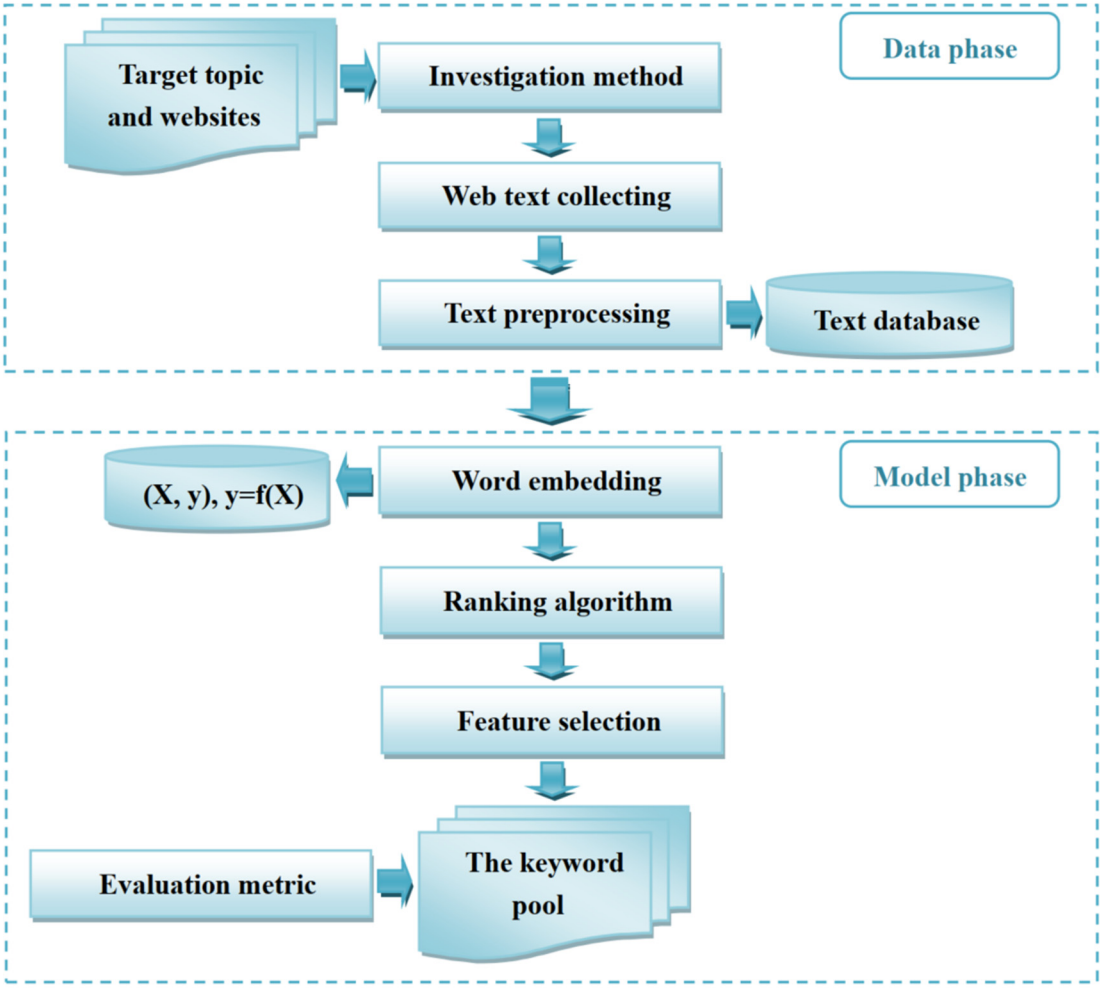


**Figure 1.** The proposed framework for generating keyword pools.

### *3.2. The Phase for Web Text Collecting and Preprocessing*

In the data phase, this is a generic phase for web text collecting and preprocessing.

Firstly, target topics and websites for the analysis need to be identified. Then comes an investigation method (e.g., random sampling investigation and typical investigation) for the web text data. Because using an appropriate investigation method can balance the efficiency of analysis and robustness of the results, while reducing costs. Next, access to web data is mainly through open data API and web page crawler. Specifically, the target website is checked firstly whether it provides an open data API, and if not, then considers web page crawling according to the website crawler protocol (e.g., robots.txt) [20]. Because the data provided by the API is already collated. This makes us more efficient, eliminating the need for excessive effort to parse web pages, and web data crawling often puts pressure on the target website, so it needs to comply with the crawler protocol [21]. Afterwards, web text collecting using the topic as a search query based on the investigation method is performed. Finally, the collected text needs to be preprocessed. Text preprocessing is to improve the efficiency and effectiveness of text mining, and generally includes two types of preprocessing at text level and word level. In detail, the text level includes mainly text cleaning (e.g., duplicate text removal, empty field removal and non-textual content removal, e.g., Hypertext Markup Language tag removal), text filtering (e.g., non-text and irrelevant text removal and text with insufficient valid characters removal), and text normalization (e.g., lowercase conversation). The word level focuses on word segmentation, part-of-speech tagging, word cleaning (e.g., stop word and low frequency word removal), word standardization (e.g., stemming or lemmatization) and spelling correction (e.g., word

spelling and grammar usage correction) [1,22]. After text preprocessing, the target text database is constructed.

### *3.3. The Phase for a Keyword Pool Generating*

In the model phase, this is a generic phase for a keyword pool generating. At first, text representation is to allow the computer to recognize and understand text data, and the preprocessed text is always represented as a word vector or matrix [23,24]. A word embedding approach, such as word to vector (Word2Vec) [25] and Bidirectional Encoder Representations from Transformers (BERT) [26], is used to train the word vectors for vectorized representation of text in low dimensions. The word vectors can then be split into a topic word vector (y) and a set of word vectors (X), denoted as (X, y). In addition, BERT does not require advance word level preprocessing such as word segmentation and word cleaning [26]. Consider the problem that if BERT is used to train word vectors, the word vectors trained for the same words in different contexts are not unique. A natural idea is to synthesize multiple word vectors of the same word (our topic) into one word vector using a dimensionality reduction method similar to principal component analysis. But this requires that our topic (the seed keyword) has unique meaning.

Next, the set of word vectors (X) are ranked by a ranking algorithm, such as SHAP-based feature importance [27] and tree-based feature importance [28], based on the topic word vector (y). Meanwhile, the regression model for X and y is constructed, which denotes as Equation (1). In addition, Shapley value can measure the importance of words, and it is effectively estimated by SHAP method according to the local agency model. The stage follows the general equation below:

$$y = f(X) \tag{1}$$

$$X_{rank} = rank(X, y) \tag{2}$$

where $y$ denotes a word vector of the topic (topic vector), $X = \{x_1, x_2, \cdots, x_n\}$ denotes the set of word vectors (except the topic vector $y$), $n$ is the number of the words (except the word of topic). $rank(X, y)$ represents a ranking algorithm used for $X$ based on $y$. $X_{rank} = \{x_{r1}, x_{r2}, \cdots, x_{rn}\}$ represents $X$ after ranking.

Three keyword generation methods have been proposed or considered. First, higher ranked word vectors are gradually introduced (The number of introduced steps should be greater than or equal to one) into the base regression model, i.e., this paper defines it as Recursive Feature Introduction (RFI) (Figure 2). Second, lower ranked word vectors are gradually eliminated (The number of eliminated steps should be greater than or equal to one) from the base regression model, i.e., Recursive Feature Elimination (RFE) [29] (Figure 3). Third, lower ranked word vectors are gradually eliminated after higher ranked word vectors are gradually introduced, and they are introduced and eliminated at the same round of training, i.e., the paper defines it as Recursive Feature Introduction and Elimination (RFIE) (Figure 4). And the average loss of the model is calculated after each round of training. In detail, the regression model, always statistical learning models, e.g., random forest (RF), support vector regression (SVR) and eXtreme Gradient Boosting (XGB), is constructed based on the word vectors and the topic vector, where the words are the input variables and the topic is the output variable. The average loss of models can be computed according to the $k$-fold cross-validation (CV) training regression models.

$$y = f(X_{rank,i}) \tag{3}$$

$$Loss_i = \frac{1}{k}\sum_{j=1}^{k} loss_j(y, f(X_{rank,i}))) \tag{4}$$

where $X_{rank,i}$ represents the first $i$ set of words vectors based on the order in $X_{rank}$, $i = 1, 2, 3, \cdots, n$. $y = f(X_{rank,i})$ is a base regression model for topic vector $y$ and the first $i$ set of words vectors $X_{rank,i}$. $loss(\cdot)$ represents a loss function, e.g., the Mean Square

Error (MSE). $k$ denotes the number of folds for CV. $Loss_i$ is the average loss of the regression model according to the $k$-fold CV training.

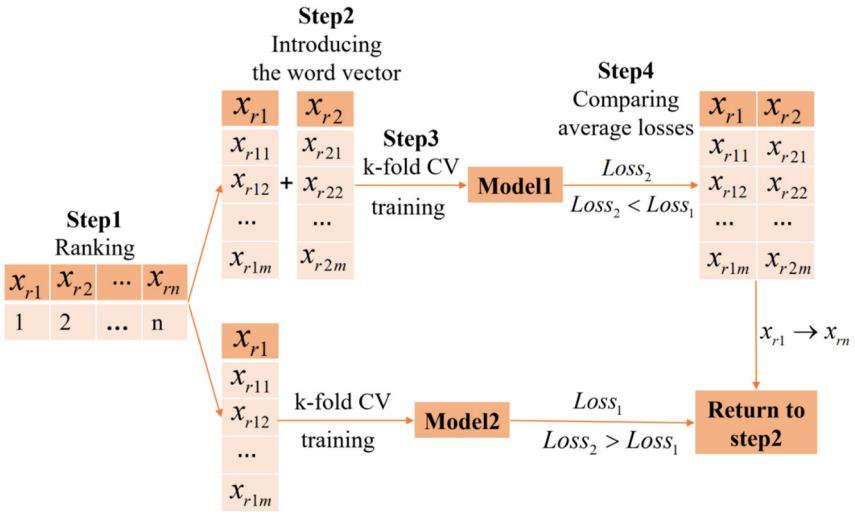


**Figure 2.** Illustration of the steps and process of RFI.

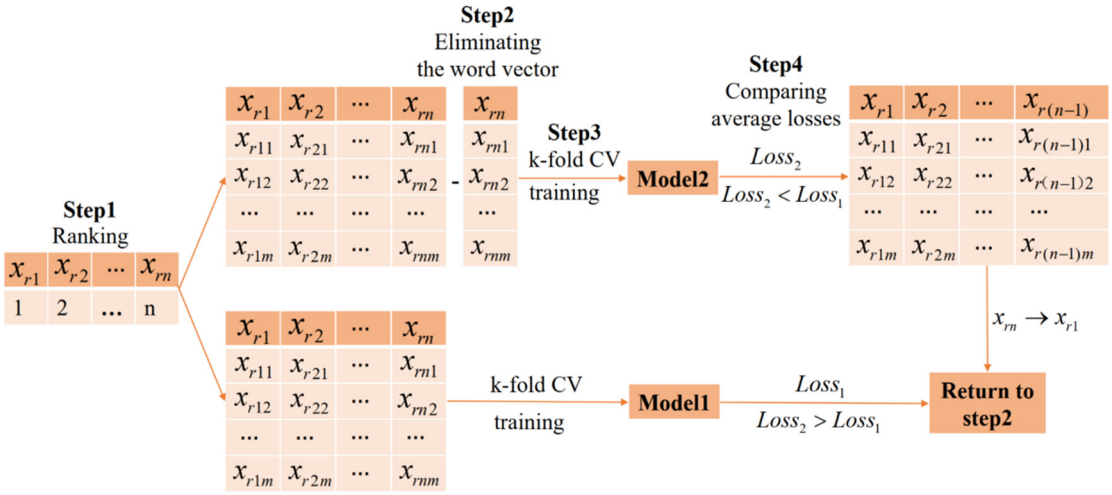


**Figure 3.** Illustration of the steps and process of RFE.

After each round of introducing or eliminating the word vector, the $Loss_{i+1}$ obtained from the training is compared with the $Loss_i$ of the previous round, if $Loss_{i+1} < Loss_i$, then continues to introduce or eliminate the new word vector, otherwise, returns to the previous round and continue to introduce or eliminate new word vectors. After all the word vectors is introduced or eliminated, the corresponding keyword pool based on the minimum average loss $Loss_l$ of the regression model according to the Equation (5) (Figures 2–4) is obtained.

$$Loss_l = \min(Loss_i) \Rightarrow f(X_{rank,l}) \Rightarrow X_{rank,l} \tag{5}$$

where $\min(Loss_i)$ denotes the minimum value of $Loss_i$, i.e., $Loss_l$.

Finally, the keyword pool needs to be further evaluated for their fit with the topic, such as conceptual coverage and relevance [8]. Further, the generated keyword pool can be used as search queries for web text crawling to provide a complete and reasonable data base for text mining.

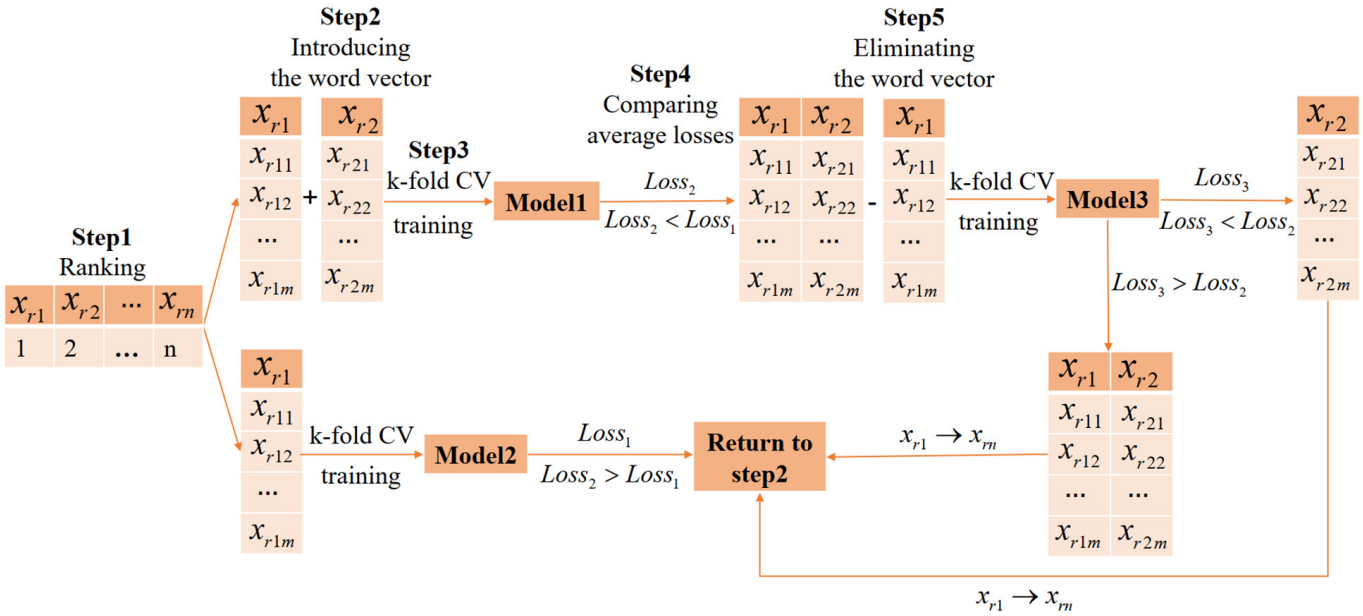


**Figure 4.** Illustration of the steps and process of RFIE.

### *3.4. Algorithms and Methods*

#### 3.4.1. Word Embedding Approaches

The word embedding approach utilizes word embeddings for text representation. It maps each word to a vector space in which the semantics of the words are salient elements, and words with similar topics are less distant from each other than words with disconnected topics [24]. There are some commonly used word embedding methods, including Word2Vec [25], Global Vectors [30], FastText [31,32], Embeddings from Language Models [33] and BERT [26]. For word embedding, considering the computational cost and model performance, the classical word embedding model Word2Vec was chosen for text representation. Word2Vec is a classical neural network-based word embedding model. It captures the contextual information of words, with better semantic representation and simpler model structure and training [25].

#### 3.4.2. Ranking Algorithm and Feature Selection

Feature ranking algorithms are used to optimize feature selection in this study. The feature ranking algorithm based on specific criteria is to compute an importance score for each feature and rank the features based on this score [34]. There are some commonly used ranking algorithms, including feature weights [35], model-based feature importance [36–38], permutation feature importance [36] and SHAP-based feature importance [39]. For feature ranking algorithm, SHAP-based and tree-based feature importance were chosen, which are currently commonly used and has better results, to rank features based on different regression models. SHAP-based and tree-based feature importance take into account the interactions between features, and SHAP-based feature significance provides a stronger ability to explain model predictions [36,39]. Feature selection is to choose the features that are important to the model, which is a very important part of feature engineering and is inseparable from the feature importance, including filters, wrappers, and embedded methods. For feature selection, the idea for our solution comes from the wrappers. The wrapper utilizes a learning machine that scores a subset of features based on their predictive power [38], i.e., its feature selection is based on model performance.

#### 3.4.3. Regression Model

In this paper, several regression models were tested to compare their differences, and testing with different models helps to get the best model and make better feature selection.

There are some commonly used regression models, including linear models [35], decision tree models [40], support vector machines [41], Bayesian models [42], non-parametric models [43], deep learning models [44] and ensemble learning [45]. Four regression models, RF, Gradient Boosting Decision Tree (GBDT), XGB and LGBM were chosen for the regression analysis. RF builds bootstrap aggregating (Bagging) ensemble using the decision tree as the base learner and further introduces random feature selection in the training process of the decision tree [36]. GBDT fits the residuals by using the negative gradient of the model over the data as an approximation of the residuals [37]. XGB belongs to ensemble learning boosting, which is an improvement of the boosting algorithm based on GBDT with the addition of a regularization term for model complexity [46]. XGB also fits the data residuals and approximates the model loss residuals with a Taylor expansion, while adding a regularization term to the loss function. LGBM fundamentals are the same as XGB, using decision trees based on learning algorithms, with the difference in the optimization of the model training speed [47].

## 4. Experiments

### *4.1. Data Sources and Preprocessing*

In our experiment, "epilepsia" was set as the topic and the search query, which was used to collect 19,888 English journal abstracts information between 1 January 2019 and 15 September 2022 based on the web crawler of PubMed. Therefore, a typical investigation was used in this experiment.

Before data preprocessing, the text data collected through the PubMed database is difficult to be analyzed directly, and the direct analysis will lead to inefficient analysis and large deviation of the results. The specific data processing steps include text level and word level. The text level mainly includes: deleting duplicated data, deleting data with empty abstract field, filtering abstracts with less than 70 valid characters according to the frequency distribution of abstract characters (following the principle of retaining more data through experiments), converting letters to lowercase, and using regular expressions to exclude abstracts with special patterns of meaninglessness (e.g., some articles will appear to have a corrected version duplicated with the uncorrected version, and the abstracts will be meaningless content). The word level consists mainly of word segmentation (breaking text into individual words so that the computer can automatically recognize the meaning of the text, i.e., each text generates a vector of word sets [48]), filtering out words with fewer than 3 valid characters, and filtering out a large number of meaningless words, which mainly include non-English text, numbers, symbols, and stop words (e.g., "this", "abstract", and "study") to improve analysis efficiency and save memory space. After data preprocessing, the experiment culminated in the construction of a database of a total of 18,267 English abstracts.

### *4.2. Experimental Details and Evaluation Metrics*

#### 4.2.1. Experimental Details

Regression models, especially machine learning models, have hyperparameters that define general features that may directly affect their performance. The most important parameters of a tree model are the number of trees and the maximum features. The higher the number of trees, the higher the performance of the model and the higher the computational cost. And if the number of trees exceeds a specific value, the prediction accuracy will no longer improve. In detail, the dataset was divided into a training dataset and a test dataset with a split ratio of 7:3 firstly. The important hyperparameters of all models went through a consistent and standard optimization process: First, given a set of broad ranges of hyperparameters, a random search was performed, and 3-fold cross-validation was used to train for 100 times, and then the approximate ranges of the optimal hyperparameters were obtained. Next, a grid search was performed based on the approximate range of hyperparameters, also trained 100 times using 3-fold cross-validation, and finally the optimal hyperparameter values were obtained. It was worth noting that in

the process of parameter optimization, the other parameters were optimized after adjusting the learning rate to the maximum first, and then optimized the learning rate separately finally, which could improve the efficiency of parameter optimization. In addition, the word vector size is set to 300, the minimum word frequency is set to 5 in Word2Vec. And the word vectors were min-max normalized after training. Other parameters in the regression models and Word2Vec are obtained through default values.

$$\hat{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{6}$$

where $x$ represents an element in a word vector, and the normalized result is denoted by $\hat{x}$.

The data phases were carried out using R (Version 4.1.3; R Foundation for Statistical Computing), and the model phase were carried out using Python (Version 3.8.5) on a PC with AMD Ryzen 7, 4800U with Radeon Graphics, 1.80 GHz, 40 GB RAM.

#### 4.2.2. Evaluation Metrics

The experiments used RFI with unranked features as a baseline and compared the differences in keyword generation between the baseline and RFI or RFE methods with ranked features (SHAP-based and tree-based ranked features) to validate the effectiveness of our framework. The regression models using 3-fold cross-validation were trained and then calculated the average loss of the models. In this case, the loss function uses MSE to evaluate the performance of the regression models to obtain the best model to help determine the best keyword pool. *MSE* is a measure of the squared difference between the true and predicted values.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{7}$$

where $N$ is the number of samples, $y_i$ is the true value of samples and $\hat{y}_i$ represents the predicted values of the model. The lower the value of MSE, the better the forecasting model's performance.

Cosine similarity is used to evaluate the similarity of two vectors by calculating the cosine of their angle, so it is often used to measure the similarity of word vectors. It can be used to evaluate the fit of a keyword pool to a topic. To more significantly compare the effects of different feature ranking algorithms, regression models, and keyword generation methods to keyword generation, the experimental process of this study tried to select the first three to ten words of each keyword pool respectively, and calculate the corresponding cumulative similarity scores in turn, and the results show that the effect is most significant when the first seven keywords are selected.

$$S = \frac{x \cdot y}{|x||y|} \tag{8}$$

where $S$ is the cosine similarity, $x$ and $y$ represent the vectors of cosine similarity to be evaluated, $|x|$ is the length of $x$ and $|y|$ is the length of $y$. The higher the cosine similarity, the more similar the two vectors are to each other.

## 5. Results Analysis

### *5.1. General Performance of the Models*

This section provides the prediction performance of the four types of regression models. Table 1 shows the keyword generation for the baseline and RFI or RFE methods with ranked features. Among them, the minimum average loss and its corresponding number of keyword generation for the different regression models are presented. And the four regression models, RF, GBDT, XGB and LGBM were compared.

**Table 1.** Evaluation of the prediction performance and keyword generation effectiveness of the different regression models in both the baseline and RFI or RFE methods based on ranked features.

| **RFI** [a] | | | | **SHAP-Based + RFI** [a] | | | |
|---|---|---|---|---|---|---|---|
| **Model** | **Keyword Number** | **MSE** [b] | **S** [c] | **Model** | **Keyword Number** | **MSE** | **S** |
| RF | 11184 | 0.0824 | 0.0357 | RF | 119 | 0.0735 | 0.0719 |
| GBDT | 6788 | 0.0815 | 0.0357 | GBDT | 210 | 0.0782 | 0.0744 |
| XGB | 5186 | 0.0932 | 0.0357 | XGB | 49 | 0.0809 | 0.1079 |
| LGBM | 11942 | 0.0835 | 0.0357 | LGBM | 118 | 0.0590 | 0.1659 |
| **Tree-Based + RFI** [a] | | | | **Tree-Based + RFE** [a] | | | |
| **Model** | **Keyword Number** | **MSE** | **S** | **Model** | **Keyword Number** | **MSE** | **S** |
| RF | 78 | 0.0682 | 0.1107 | RF | 8162 | 0.0813 | 0.1346 |
| GBDT | 27 | 0.0776 | 0.0834 | GBDT | 14459 | 0.0802 | 0.0585 |
| XGB | 90 | 0.0662 | 0.0740 | XGB | 317 | 0.1189 | 0.0740 |
| LGBM | 58 | 0.0442 | 0.1746 | LGBM | 158 | 0.0886 | 0.1746 |

[a] RFI denotes the baseline, i.e., RFI with unranked features; SHAP-based + RFI is RFI with SHAP-based ranked features; Tree-based + RFI denotes RFI with tree-based ranked features; Tree-based + RFE represents RFE with tree-based ranked features. [b] MSE represents the minimum average loss of the regression model. [c] S represents the average similarity score of the first seven keywords in the keyword generation results.

Several conclusions can be drawn from the results in Table 1. When using RFI methods to generate keywords, regression models using both SHAP-based + RFI and Tree-based + RFI have significantly smaller MSE than the baseline model, better prediction performance, and more refined pools of keywords generated. Comparing the different feature ranking algorithms, the regression model using Tree-based + RFI has significantly better prediction performance than SHAP-based + RFI. Among the regression models using SHAP-based + RFI and Tree-based + RFI, the prediction performance of LGBM is significantly better than that of RF, GBDT, and XGB. Compared to other models, it was also found in the experiments that the MSE change of LGBM was more affected by the feature ranking algorithms. Specifically, RF, XGB, and LGBM have the smallest MSE and best prediction performance when using Tree-based + RFI, and GBDT has best prediction performance when using SHAP-based + RFI. Considering the different keyword generation methods, the regression model using the Tree-based + RFI significantly outperforms the one using the Tree-based + RFE, and the keyword pools generated by the former are also more refined. Specifically, GBDT prediction performance is best with using the baseline and Tree-based + RFE, and LGBM prediction performance was is best with using SHAP-based + RFI and Tree-based + RFI.

### *5.2. Evaluation of the Keyword Pools*

This study evaluated the performance of the keyword pools generated based on this framework and the baseline in terms of similarity. A higher average similarity score indicates that the generated keyword pool is more similar to the topic. By calculating the average similarity scores of the top seven ranked keywords in all the keyword pools (Table 1), the study found that the keyword pools generated by SHAP-based + RFI, Tree-based + RFI, and Tree-based + RFE have significantly higher average similarity scores than the keyword pools generated by the baseline. Considering the different keyword generation methods of RFI and RFE, the average similarity score of keyword pools generated by Tree-based + RFI are not significantly different from that by Tree-based + RFE. Comparing the different regression models, LGBM has significantly higher average similarity scores than RF, GBDT, and XBG for generating keyword pools regardless of whether RFI or RFE with SHAP-based or tree-based ranked features, which indicates that LGBM performs better than the other models in generating keywords that are more similar to the topic.

Based on the average similarity score for each keyword, a graphical representation of the cumulative similarity scores of the top seven ranked keywords is given in Figure 5, where the horizontal axis represents the number of keywords and the vertical axis shows the cumulative similarity scores. In Figure 5, the cumulative advantage of the four regression models using SHAP-based + RFI, Tree-based + RFI, and Tree-based + RFE over the baseline model becomes more and more significant as the number of keywords increases. Among them, considering the different ranking algorithms and keyword generation methods, the cumulative curves of RF, GBDT, and LGBM using Tree-based + RFI and Tree-based + RFE are all at the top respectively, indicating that these models have the most significant cumulative advantage using Tree-based + RFI and Tree-based + RFE, whereas XGB has the most significant cumulative advantage using SHAP-based + RFI. LGBM has the significant cumulative advantage for all the regression models regardless of whether the use of RFI or RFE with SHAP-based or Tree-based ranked features.

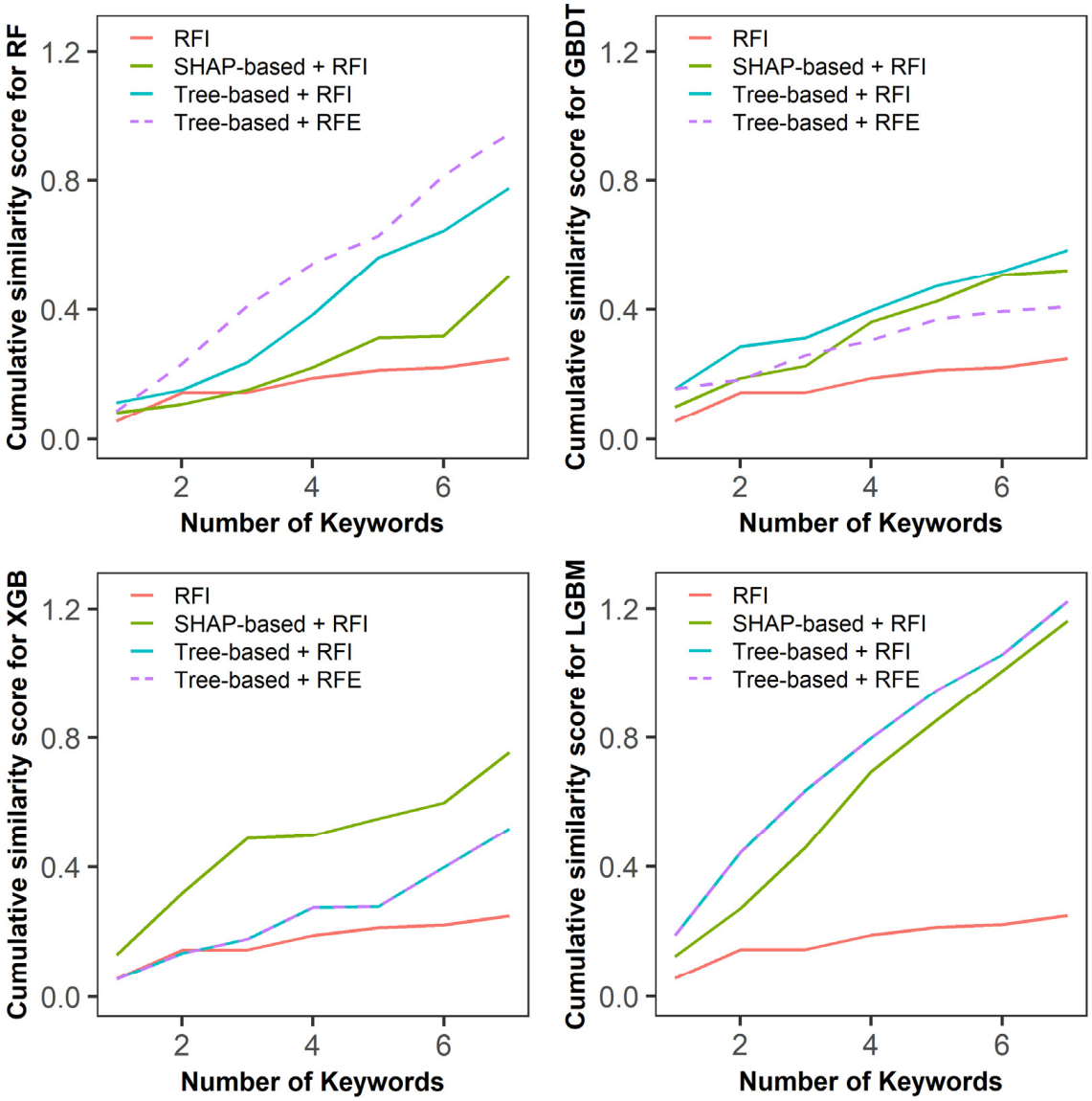


**Figure 5.** Cumulative similarity scores of keywords obtained by different keyword generation methods and regression models.

In addition, the study found that the average similarity scores of XGB were the same and their cumulative curves overlapped when using Tree-based + RFI and Tree-based + RFE. LGBM also produced the same results as XGB. These show that XGB/LGBM performs

equally well in terms of average similarity score and cumulative similarity score when keywords are generated by RFI and RFE methods. This is possible because both Tree-based + RFI and Tree-based + RFE use the same feature ranking algorithm, and in the step-by-step process of generating (introducing or eliminating) keywords, the word vectors with higher rankings (importance) have a high probability of being generated, so the first few keywords generated may be the same.

## 6. Conclusions and Future Work

A refined keyword pool with high similarity to the topic and some extensibility is crucial for web text data collecting, and even affects the subsequent text mining analysis. However, in the field of text mining, there is little discussion on keywords as search queries, either lacking an objective generation method and evaluation system, or underutilizing the sample semantic information. Therefore, this paper proposed a keyword generation framework that integrates sample and semantic information, and also proposed two keyword generation methods, RFI and RFIE, which to a certain extent bridges the gap in keyword generation-related research, and provides certain research ideas for expanding the research sample size while ensuring the coverage of topics. The core of the framework considers both feature ranking (word order) and model performance, and constructs a regression model on a topic vector and word vectors based on word embedding.

Our study shows that when generating keywords using RFI method, the regression model using ranked features has better prediction performance than the baseline model (the one using unranked features), generates a pool of keywords with higher average similarity scores and cumulative similarity scores, and a more refined number of keywords, and that the regression model using tree-based ranked features has significantly better prediction performance than the one using SHAP-based ranked features. As far as the keyword generation methods are concerned, the prediction performance of the regression models using tree-based + RFI is significantly better than that using tree-based + RFE, but the average similarity scores and cumulative similarity scores of the (higher ranked) keywords generated using the two methods present different performances depending on the regression model, and the performance is the same on the XBG/LGBM. Compared to the baseline, using the four regression models (RF, GBDT, XGB and LGBM) based on SHAP + RFI, Tree + RFI and Tree + RFE not only showed significantly higher average similarity scores, but also an increasingly significant cumulative advantage. Taken together, the keyword pool generated by LGBM has higher average similarity scores and cumulative similarity scores compared to RF, GBDT, and XGB, and it has better prediction performance when using ranked features, but its performance is more affected by the feature ranking algorithm, which suggests that LGBM outperforms the other models in acquiring keywords that are more similar to the topic. In conclusion, LGBM using SHAP-based + RFI method to generate keywords performs best Therefore, other studies can learn from this effective approach to generate a refined keyword pool with higher similarity to the topic and some extensibility based on the framework proposed in this paper, and use the keyword pool as a search query reference for web text collecting.

This study has some limitations. First, limited to the computing platform, the study only used the tree-based ensemble models in the experiments, while the kernel function models like SVR [41] requires a higher performance computing platform. And for RFIE keyword generation method, the experiment cannot be implemented in the short term due to the huge amount of computation. Nevertheless, this framework is still generalizable and the computing platform will be upgraded at a later stage to further validate the framework. In addition, the keywords in the study are based on words to generate keyword pools. The form of phrases rather than words will be considered for generating keyword pools, but the use of phrases will greatly increase the computation amount.

In future work, in addition to expanding word embedding models, feature ranking algorithms, keyword generation methods, and regression models, covering more types

of topics and comparing the differences in the effectiveness of generating keyword pools between different topics are our next research direction.

**Author Contributions:** X.W.: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing—original draft, Writing—review & editing, Visualization. C.F.: Conceptualization, Methodology, Writing—review & editing. Q.L.: Validation, Formal analysis, Writing—review & editing, Resources, Project administration. J.Z.: Conceptualization, Writing—review & editing, Resources, Supervision, Project administration, Funding acquisition. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Major project of National Social Science Fund of China [grant numbers 20&ZD137].

**Data Availability Statement:** Data will be made available on request.

**Acknowledgments:** The authors are grateful to all study participants.

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

## References

1. Xie, X.; Fu, Y.; Jin, H.; Zhao, Y.; Cao, W. A novel text mining approach for scholar information extraction from web content in Chinese. *Future Gener. Comput. Syst.* **2020**, *111*, 859–872. [CrossRef]
2. Zhang, Y.; Chen, K.; Weng, Y.; Chen, Z.; Zhang, J.; Hubbard, R. An Intelligent Early Warning System of Analyzing Twitter Data Using Machine Learning on COVID-19 Surveillance in the US. *Expert Syst. Appl.* **2022**, *198*, 116882. [CrossRef]
3. Hung, M.; Lauren, E.; Hon, E.S.; Birmingham, W.C.; Xu, J.; Su, S.; Hon, S.D.; Park, J.; Dang, P.; Lipsky, M.S. Social network analysis of COVID-19 sentiments: Application of artificial intelligence. *J. Med. Internet Res.* **2020**, *22*, e22590. [CrossRef]
4. Ozbay, F.A.; Alatas, B. Fake news detection within online social media using supervised artificial intelligence algorithms. *Phys. A Stat. Mech. Its Appl.* **2020**, *540*, 123174. [CrossRef]
5. Akbari Torkestani, J. An adaptive focused Web crawling algorithm based on learning automata. *Appl. Intell.* **2012**, *37*, 586–601. [CrossRef]
6. Batsakis, S.; Petrakis, E.G.; Milios, E. Improving the performance of focused web crawlers. *Data Knowl. Eng.* **2009**, *68*, 1001–1013. [CrossRef]
7. Kaur, S.; Singh, A.; Geetha, G.; Masud, M.; Alzain, M.A. SmartCrawler: A Three-Stage Ranking Based Web Crawler for Harvesting Hidden Web Sources. *CMC-Comput. Mater. Contin.* **2021**, *69*, 2933–2948. [CrossRef]
8. Nie, H.; Yang, Y.; Zeng, D. Keyword generation for sponsored search advertising: Balancing coverage and relevance. *IEEE Intell. Syst.* **2019**, *34*, 14–24. [CrossRef]
9. Joshi, A.; Motwani, R. Keyword generation for search engine advertising. In Proceedings of the Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06), Hong Kong, China, 18–22 December 2006; pp. 490–496. [CrossRef]
10. Cronin, J.; Mao, Y.; Menchen-Trevino, E. Connecting During a Government Shutdown: Networked Care and the Temporal Aspects of Social Media Activism. *Soc. Media+ Soc.* **2022**, *8*, 20563051211069054. [CrossRef]
11. Michalko, D.; Plichtová, J.; Šestáková, A. Network analysis approach for exploring dementia representations in the Slovak media. *Dementia* **2022**, *21*, 781–793. [CrossRef] [PubMed]
12. Zhao, F.; Skums, P.; Zelikovsky, A.; Sevigny, E.L.; Swahn, M.H.; Strasser, S.M.; Huang, Y.; Wu, Y. Computational approaches to detect illicit drug ads and find vendor communities within social media platforms. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2020**, *19*, 180–191. [CrossRef]
13. Wu, X.; Wang, W.; Li, Q.; Peng, Z.; Zhu, J. Current Situation with Organ Donation and Transplantation in China: Application of Machine Learning. *Transplant. Proc.* **2022**, *54*, 1711–1723. [CrossRef] [PubMed]
14. Chen, B.; Chen, X.; Pan, J.; Liu, K.; Xie, B.; Wang, W.; Peng, Y.; Wang, F.; Li, N.; Jiang, J. Dissemination and refutation of rumors during the COVID-19 outbreak in China: Infodemiology study. *J. Med. Internet Res.* **2021**, *23*, e22427. [CrossRef]
15. Bhatt, P.; Vemprala, N.; Valecha, R.; Hariharan, G.; Rao, H.R. User Privacy, Surveillance and Public Health during COVID-19–An Examination of Twitter verse. *Inf. Syst. Front.* **2022**, *25*, 1667–1682. [CrossRef] [PubMed]
16. Barchiesi, M.A.; Colladon, A.F. Big data and big values: When companies need to rethink themselves. *J. Bus. Res.* **2021**, *129*, 714–722. [CrossRef]
17. Chen, Y.; Xue, G.R.; Yu, Y. Advertising keyword suggestion based on concept hierarchy. In Proceedings of the 2008 International Conference on Web Search and Data Mining, Palo Alto, CA, USA, 11–12 February 2008; pp. 251–260. [CrossRef]
18. Zhang, W.; Wang, D.; Xue, G.R.; Zha, H. Advertising keywords recommendation for short-text web pages using Wikipedia. *ACM Trans. Intell. Syst. Technol. (TIST)* **2012**, *3*, 1–25. [CrossRef]
19. Zhou, H.; Huang, M.; Mao, Y.; Zhu, C.; Shu, P.; Zhu, X. Domain-constrained advertising keyword generation. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2448–2459. [CrossRef]
20. Martin-Galan, B.; Hernandez-Perez, T.; Rodriguez-Mateos, D.; Pena-Gil, D. The use of robots.txt and sitemaps in the Spanish public administration. *Inf. Prof.* **2009**, *18*, 625–632. [CrossRef]

21. Wen, Y.F.; Hung, K.Y.; Hwang, Y.T.; Lin, Y.S.F. Sports lottery game prediction system development and evaluation on social networks. *Internet Res.* **2016**, *26*, 758–788. [CrossRef]
22. Hickman, L.; Thapa, S.; Tay, L.; Cao, M.; Srinivasan, P. Text preprocessing for text mining in organizational research: Review and recommendations. *Organ. Res. Methods* **2022**, *25*, 114–146. [CrossRef]
23. Wang, H.; Liu, Z.; Xu, Y.; Wei, X.; Wang, L. Short text mining framework with specific design for operation and maintenance of power equipment. *CSEE J. Power Energy Syst.* **2020**, *7*, 1267–1277. [CrossRef]
24. Rahimi, Z.; Homayounpour, M.M. Tens-embedding: A tensor-based document embedding method. *Expert Syst. Appl.* **2020**, *162*, 113770. [CrossRef]
25. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781. [CrossRef]
26. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language under-standing. *arXiv* **2018**, arXiv:1810.04805. [CrossRef]
27. Shapley, L.S. A Value for n-Person Games. In *Contributions to the Theory of Games*; Princeton University Press: Princeton, NJ, USA, 1953; Volume 2, pp. 307–317. [CrossRef]
28. Medelyan, O.; Frank, E.; Witten, I.H. Human-Competitive Tagging Using Automatic Keyphrase Extraction. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–7 August 2009; pp. 1318–1327. Available online: https://dl.acm.org/doi/10.5555/1699648.1699678 (accessed on 1 September 2023).
29. Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. Gene Selection for Cancer Classification using Support Vector Machines. *Mach. Learn.* **2002**, *46*, 389–422. [CrossRef]
30. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543. [CrossRef]
31. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. *arXiv* **2016**, arXiv:1607.01759. [CrossRef]
32. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist* **2017**, *5*, 135–146. [CrossRef]
33. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word represen-tations. *arXiv* **2018**, arXiv:1802.05365. [CrossRef]
34. Janaki, M.; Geethalakshmi, S.N. A review of swarm intelligence-based feature selection methods and its application. In *International Conference on Soft Computing for Security Applications (ICSCS), Advances in Intelligent Systems and Computing*; Springer: Singapore, 2023; Volume 1428, pp. 435–447. [CrossRef]
35. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer Series in Statistics; Springer: New York, NY, USA, 2009.
36. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
37. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
38. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
39. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.I. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2020**, *2*, 56–67. [CrossRef] [PubMed]
40. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; CRC Press: Boca Raton, FL, USA, 1984.
41. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: New York, NY, USA, 1999.
42. Gelman, A.; Carlin, J.B.; Stern, H.S.; Dunson, D.B.; Vehtari, A.; Rubin, D.B. *Bayesian Data Analysis*, 3rd ed.; CRC Press: Boca Raton, FL, USA, 2013.
43. Wasserman, L. *All of Nonparametric Statistics*; Springer Science & Business Media: New York, NY, USA, 2006.
44. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning (No. 2)*; MIT Press: Cambridge, UK, 2016.
45. Dietterich, T.G. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems. MCS 2000*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1857. [CrossRef]
46. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–16 August 2016; pp. 785–794. [CrossRef]
47. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 3146–3154. Available online: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf (accessed on 1 September 2023).
48. Salton, G.; Wong, A.; Yang, C.S. A vector space model for automatic indexing. *Commun. ACM* **1975**, *18*, 613–620. [CrossRef]