symmetry

MDPI

Review

A Review of Unsupervised Keyphrase Extraction Methods Using Within-Collection Resources

Chengyu Sun, Liang Hu, Shuai Li, Tuohang Li, Hongtu Li and Ling Chi *

College of Computer Science and Technology, Jilin Unversity, Chaoyang District, Changchun 130012, China;
cysun20@mails.jlu.edu.cn (C.S.); hul@jlu.edu.cn (L.H.); shuaili18@mails.jlu.edu.cn (S.L.);
lith19@mails.jlu.edu.cn (T.L.); lihongtu@jlu.edu.cn (H.L.)
* Correspondence: chiling@jlu.edu.cn

Received: 22 October 2020; Accepted: 9 November 2020; Published: 12 November 2020

check for updates

Abstract: An essential part of a text generation task is to extract critical information from the text. People usually obtain critical information in the text via manual extraction; however, the asymmetry between the ability to process information manually and the speed of information growth makes it impossible. This problem can be solved by automatic keyphrase extraction. In this paper, the mainstream unsupervised methods to extract keyphrases are summarized, and we analyze in detail the reasons for the differences in the performance of methods then provided some solutions.

Keywords: keyphrase extraction; unsupervised method; feature selection

1. Introduction

Under the background of the continuous development of the information age, the content based on words grows exponentially, making it more challenging to manage this large-scale information. This information could be processed manually in the past. However, now, it is impossible because of the asymmetry between the amount of data and the ability to process information manually, which exemplifies the efforts to handle the current data scales, thereby promoting the development of automatic key sentence and keyphrase extraction methods that use the mighty computing power of computers to replace the manual labor [1]. Keyphrase extraction and key sentence extraction are two important subtasks in the text generation task [2–5]. Among them, the key sentence extraction task separates the most important part of a text and combines it in a specific way to an abstract that can express the text's main content while retaining the readability [6]. The main task of keyphrase extraction is to identify a single word or phrase representing the text's main content [7]. The extracted results are called keyphrases, the most common of which include the keyword in the abstracts of academic papers, representing the core content that the author wants to express. As the concise expression of an article's main idea, keyphrase makes the information easy to be managed, classified, and retrieved [8]. At present, keyphrase extraction is widely used in many fields, such as natural language processing (NLP), information retrieval (IR) [9–12], opinion mining [13–15], document indexing [16], and document classification [17].

Keyphrase extraction is divided into supervised methods and unsupervised methods based on the training set. The difference between them is whether there is a labeled training set in the learning process. Among them, the supervised method [18] transforms the keyphrase extraction task into a classification problem [19,20] or regression problem [21]. It trains the model on the labeled training set and uses the trained model to determine whether a candidate word in a text is a keyphrase. For example, KEA (Automatic keyphrase extraction) [19] determines whether a candidate word is a keyphrase by calculating the TF-IDF (Term Frequency–Inverse Document Frequency) [22] value of each candidate word and the location where it first appears in the text and inputs these two values into the Naive Bayes classifier. Generally, the supervised method is superior to the unsupervised method [23]. However, compared with the past, the explosive growth of all kinds of information makes the types and quantity of information increase significantly, and the supervised method requires many labeled training sets, thus it require large amounts of manual labor [24]. Moreover, there are no labeled datasets that can serve as references in many fields, especially in some languages that are not well known by human beings, such as the translation tasks of hieroglyphs and cuneiform characters, which makes unsupervised methods without human intervention essential.

Based on the features of the unsupervised keyphrase extraction methods selected by researchers, unsupervised methods can be divided into the statistics-based method, graph-based method, topic-based method, language model-based method, and these methods can be classified into two schools: the linguistic school and the statistical school. The first school mainly extracts keyphrases by analyzing texts using linguistic methods, among which the most common method is to analyze the topic distribution of articles, such as KeyCluster [25] and CommunityCluster [26]. The statistical school mainly analyzes an article's probability features such as KP-Miner [27] and YAKE [28] based on TF-IDF, TextRank [29], or SingleRank [30]. The linguistic school and statistical school have been influencing and promoting each other. As time has passed, researchers have proposed new methods to cross-utilize the two schools' knowledge, such as TopicRank based on clustering (linguistic school) and graphs (statistical school).

In the above discussion, we divide keyphrase extraction into the linguistic school and the statistical school. We continue this classification method to divide commonly used metrics, features that affect keyphrase extraction, and mainstream unsupervised keyphrase extraction methods, making the structure and development path of the entire field look clear.

This paper aims to introduce the mainstream unsupervised learning methods, which are reflected in [26,31], but we have done other work as follows:

- Based on the characteristics of different methods, combined with the human language habit, the reasons for the performance differences between methods are analyzed in detail (Section 5.1).
- In keyphrase extraction, the characteristics of the datasets directly affect the performance of the methods, so we analyze how different datasets affect method performance (Section 5.2).
- We analyze the reasons for the limitations of the keyphrase methods and propose corresponding solutions, which will help the following researchers to explore further (Section 6).

The remainder of this paper is organized as follows. Section 2 introduces some preliminary knowledge, including the datasets (Section 2.1) and evaluation metrics (Section 2.2) that are commonly used in the automatic keyphrase extraction field, the features affecting keyphrase extraction (Section 2.3), and how to use these features for keyphrase extraction (Section 2.4). Section 3 mainly introduces several types of unsupervised methods for keyphrase extraction (Section 3.1), which are divided into statistics-based methods (Section 3.2), graph-based methods (Section 3.3), topic-based methods (Section 3.4), and language model-based methods (Section 3.5). Section 4 is the experimental results, and we analyze the reasons for the differences in performance of methods based on human language habits in Section 5. In Section 6, we analyze the limitations of keyphrase extraction methods and provided some solutions. Finally, this paper is summarized in Section 7.

2. Datasets, Evaluation Metrics and Features

2.1. What Datasets Are There in the Keyphrase Extraction Field?

An unsupervised keyphrase extraction system can be applied to many datasets for testing, such as the full-text of a paper, the abstract of a paper, news, web page, and email. In this paper, the names, types, number of texts (Docs), contributors, number of tokens per text, language, and annotation (annotation for gold keyphrases are performed by authors (A), readers (R), editors (E), or professional indexers (I)) of multiple datasets are sorted out in detail, as shown in Table 1.

Table 1. Evaluation datasets grouped by their type.

Type	Dataset	Contributor	Tokens/Doc	Docs	Keyphrases/Doc	Language	Annotation
Full-text papers	ACM	Krapivin et al. [32]	2∼12k	2304	6	English	A
	Citeulike-180	Medelyan et al. [33]		181	5	English	A+R
	CSTR	Witten et al. [19]		630	-	English	A
	SemEval-2010	Kim et al. [23]		283	15	English	A+R
	NUS	Nguyen and Kan [34]		211	11	English	A+R
	PubMed	Schutz [35]		1320	5	English	A
Papers abstracts	Inspec	Hulth [36]	100∼200	2000	10	English	I
	KDD	Gollapalli et al. [37]		755	4	English	A
	WWW	Gollapalli et al. [37]		1330	5	English	A
	TALN	Boudin [38]		641	4	English	A
	TermLTH-Eval	Bougouin [39]		400	12	English	I
News	DUC-2001	Wan and Xiao [40]	300∼850	308	10	English	R
	500N-KPCrowd	Marujo et al. [41]		500	46	English	R
	110-PT-BN-KP	Marujo et al. [42]		110	28	Portuguese	R
	Wikinews	Bougouin et al. [7]		100	10	French	R
Web pages	Blogs	Grineva et al. [43]	500∼1k	252	8	English	R
	-	Hammouda et al. [44]		312	-	English	-

2.2. *What Are the Evaluation Metrics in the Keyphrase Extraction Field?*

It is not an easy task to design an evaluation metric that can reflect an algorithm's advantages and disadvantages. Since an evaluation metric may only evaluate one aspect of the algorithm, multiple metrics can more precisely and comprehensively evaluate an algorithm. For example, researchers usually use precision, recall, and F-score to evaluate a method from multiple perspectives. In this section, some standard evaluation metrics are introduced and divided into statistics-based and linguistics-based ones.

2.2.1. Statistics-Based Metrics

Statistics-based evaluation metrics analyze the performance of a method by calculating the proportion of the number of various keyphrases, such as the number of extracted keyphrases, correct keyphrases, wrong keyphrases, and manually assigned keyphrases. Standard statistics-based metrics include precision, recall, and F1-score.

Precision:

It represents the number of real keyphrases in the extracted keyphrases, reflecting the accuracy of the keyphrases output by the algorithm.

$$precison = \frac{tp}{tp+fp} = \frac{correct\ keyphrases}{extracted\ keyphrases} \tag{1}$$

Here, tp represents true positives, i.e., the number of keyphrases that are correctly extracted, and fp represents false positives, i.e., the number of keyphrases that are incorrectly extracted.

Recall:

It represents the number of extracted keyphrases among the real keyphrases, reflecting the comprehensiveness of the keyphrases output by the algorithm.

$$recall = \frac{tp}{tp+fn} = \frac{correctly\ matched\ keyphrases}{assigned\ keyphrases} \tag{2}$$

Here, fn represents false negatives, which are the keyphrases that are not correctly extracted.

Fα-score:

The precision and recall interact with each other. In an ideal situation, they are both high, but, in general, when precision is high, recall is low, and vice versa. The F-score is formed by combining precision and recall.

$$F\alpha\text{-}score = \frac{(\alpha^2 + 1) \cdot precision \cdot recall}{\alpha^2 \cdot precision + recall} \tag{3}$$

When α = 1, it is the $F1$-*score*.

2.2.2. Linguistics-Based Metrics

The above evaluation metrics are based on the assumption that keyphrases are mutually independent, but, based on human language habits, we hope that the more essential keyphrases should be ranked higher.

The following three evaluation metrics can reflect the order features between the keyphrases output by an algorithm.

Mean Reciprocal Rank (MRR):

In MRR [45], $rank_d$ is denoted as the rank of the first correct keyphrase with all extracted keyphrases, D is the document set for keyphrase extraction, and d is a specific document.

$$MRR = \frac{\sum\limits_{d \in D} \frac{1}{rank_d}}{|D|} \tag{4}$$

Mean Average Precision (MAP):

The MAP takes the ordering of a particular returned list of keyphrases into account. The average precision (AP) is defined as follows:

$$AP = \frac{\sum\limits_{\text{n=1}}^{|N|} P(n) gd(n)}{|LN|} \tag{5}$$

where $|N|$ is the length of the list, $|LN|$ is the number of relevant items, $P(n)$ is the precision, and $gd(n)$ equals one if the nth item is gold keyphrase and 0 otherwise. By averaging AP over a set of n documents, the Mean Average Precision (MAP) is defined as follows:

$$MAP = \frac{1}{n} \sum_{i=1}^{n} AP_i \tag{6}$$

where AP_i is the average precision of the extracted keyphrases list.

Binary Preference Measure (Bpref):

The $Bpref$ [46] represents the number of correct keyphrases in front of incorrect keyphrases extracted by the algorithm. Its definition is as follows:

$$Bpref = \frac{1}{C} \sum_{c \in C} 1 - \frac{|I|}{M} \tag{7}$$

where C represents the number of correct keyphrases, M represents the number of all extracted keyphrases, and I represents the number of correct keyphrases in front of incorrect keyphrases.

We organize all the formulas in Table 2.

Table 2. Formulas for all evaluation metrics.

Evaluation Metrcis	precision	$precison = \frac{tp}{tp+fp} = \frac{thenumberofcorrectkeyphrase}{thenumberofextractedkeyphrase}$
	recall	$recall = \frac{tp}{tp+fn} = \frac{the\ number\ of\ correctly\ matched\ keyphrase}{the\ number\ of\ assigned\ keyphrase}$
	F-socre	$F\alpha - score = \frac{(\alpha^2+1)\cdot precision \cdot recall}{\alpha^2 \cdot precision + recall}$
	MRR	$MRR = \frac{\sum_{d\in D}\frac{1}{rank_d}}{\lvert D\rvert}$
	AP	$AP = \frac{\sum_{\mathrm{n=1}}^{\lvert N\rvert} P(n)gd(n)}{\lvert LN\rvert}$
	MAP	$MAP = \frac{1}{n}\sum_{i=1}^{n} AP_i$
	Bpref	$Bpref = \frac{1}{C}\sum_{c\in C} 1 - \frac{\lvert I\rvert}{M}$

2.3. What Are the Features that Affect Keyphrase Extraction?

Many features affect keyphrase extraction methods performance, and these features are divided into linguistic-based features and statistical-based features.

2.3.1. Linguistic-Based Features

Topic distribution:

The locations where keyphrases appear are often not fixed in different text types and are affected by the distribution of topics since human language habits determine new keyphrases will appear whenever a new topic appears [8]. In academic papers and scientific articles, there is only one topic in the whole text, and so the keyphrases usually appear at the beginning and the end of a text [33]. However, most texts contain multiple topics, such as news and web pages, so new keyphrases will appear when the topic changes. To extract keyphrases from multi-topics articles, researchers have introduced clustering methods, such as Latent Dirichlet Allocation (LDA) [47], KeyCluster [25], the Topical PageRank (TPR) [24], CommunityCluster [26], and the topic-sensitive Topical PageRank (tsTPR) [48]. These methods are described in detail in Section 3.

Topic correlation:

For texts such as academic papers, the text's keyphrases are typically related to the others, so the correlation between texts can be used when extracting keyphrases [40]. However, this observation does not necessarily hold for emails or chats because there are no restrictions on the topics discussed between people, so it is difficult to use the relationship between the texts to extract keyphrases, and further increase the difficulty of the keyphrase extraction task.

2.3.2. Statistical-Based Features

Keyphrase density:

The concept of keyphrase density is proposed and defined as the ratio of the frequency of a keyphrase's occurrence to the total number of words in a text. To improve the algorithm's performance, we need to preprocess the document before calculating the keyphrase density, that is, delete the function words and restore the remaining words to their root patterns. The keyphrase density is usually related to the document's length, while the average length of the document in different datasets is often different. For example, there are 300 documents in the DUC-2001 dataset, with an average of 847 words in each document, and its keyphrase density is 0.56%. There are 1330 documents in the WWW dataset, with an average of 163 words in each document, which keyphrase density of 0.87%. According to the relevant experimental experience, the longer the document length is, the more difficult it is to extract keyphrases [26]. Therefore, the higher the keyphrase density in a document is, the easier it is to extract keyphrases because a lower keyphrase density means that there are relatively few keyphrases, and the document is relatively long, which makes it more difficult to extract real keyphrases.

Lexical density:

The lexical density is used to express the structure and complexity of human language [48]. The definition of a lexical density is the ratio of the number of lexical words, which are simply nouns, adjectives, verbs, and adverbs in the document, to the total number of words in the document [49]. Lexical words give a text its meaning and provide information regarding what the text is about. In the keyphrase extraction task, lexical words are usually used as candidate keyphrase, so, when there are more lexical words in a text (larger lexical density), we need to select the real keywords from more candidate words, which increases the difficulty.

Keyphrase density and lexical density are used to reflect the features of datasets. Their difference is that keyphrase density reflects the frequency of the keyphrases, while the lexical density reflects the richness of text semantics.

Structural features:

The difficulty of keyphrase extraction will be reduced by their fixed structures. In texts with fixed formats, such as scientific research papers, which generally include an abstract, introduction, related work, experiment, and conclusion, keyphrases often appear at fixed positions such as the abstract and conclusion [19,33]. Simultaneously, it is more challenging to extract keyphrases in texts without a fixed format, such as news, blogs, and email [23].

2.4. How to Use These Features for Keyphrase Extraction?

In Section 2.3, the features that affect keyphrase extraction are introduced, and we show how researchers use these features to complete the keyphrase extraction task. The explanation is divided into linguistic-based and statistical-based sections.

2.4.1. Linguistic-Based

Topic distribution:

As mentioned above, the topic distribution has an impact on the difficulty of keyphrase extraction. Researchers expect to extract keyphrases that can cover all topics of a given document; therefore, they take the topic distribution into account and generally use Latent Dirichlet Allocation (LDA) [47] or a clustering method to detect the topic distribution. Tthese methods are described in detail in Section 3.

Syntactic features:

It can be seen that keyphrases are generally composed of lexical words; therefore, grammar patterns can be set to filter candidate words, such as nouns or adjectives plus nouns [50]; thus, in the keyphrase extraction task, the first step is to delete the non-lexical word and select keyphrase from the remaining words.

2.4.2. Statistical-Based

Frequency of words:

Generally, if a lexical word appears more frequently in a text and less frequently in another text, it can better represent the document's critical information. Based on this finding, researchers proposed the TF-IDF [22], where TF is the term frequency, representing the frequency of a candidate word in a document, and IDF is the inverse document frequency, representing the frequency of the candidate word in other documents.

Distance of words:

Generally, if a word appears at the top of a document, it is more likely to be a keyphrase [34]. Based on this finding, researchers took the location information as a feature defined as the distance of the first occurrence of a word in a document, and the length of documents is usually used to regularize each word's location information.

Structural features:

As stated in Section 2.3.2, for texts with a fixed format, such as scientific research papers, if a word often appears in abstract and introduction, it is more likely to be a keyphrase.

3. Unsupervised Keyphrase Extraction Methods

This section introduces the classification of unsupervised keyphrase extraction methods (Section 3.1).

We introduce various types of unsupervised methods following the chronological order of the publication of papers and show how these research works are optimized from generation to generation (Sections 3.2–3.5).

3.1. Classification of Unsupervised Keyphrase Extraction Methods

The mainstream unsupervised keyphrase extraction methods are divided into four categories: statistics-based method, graph-based method, topic-based method, and language model-based method. The methods covered in this article are summarized in Figure 1.

The mainstream unsupervised methods usually preprocess documents when performing keyphrase extraction task. Because the keyphrases are the lexical words (nouns, adjectives, verbs, and adverbs), deleting other words except lexical words in the document is necessary. Since some words have different forms but have similar meanings (such as play and playing), they are restored to their root forms. Finally, the remaining words are treated as candidate keyphrases, where the real keyphrases are extracted. The unsupervised method described below does not introduce this step.

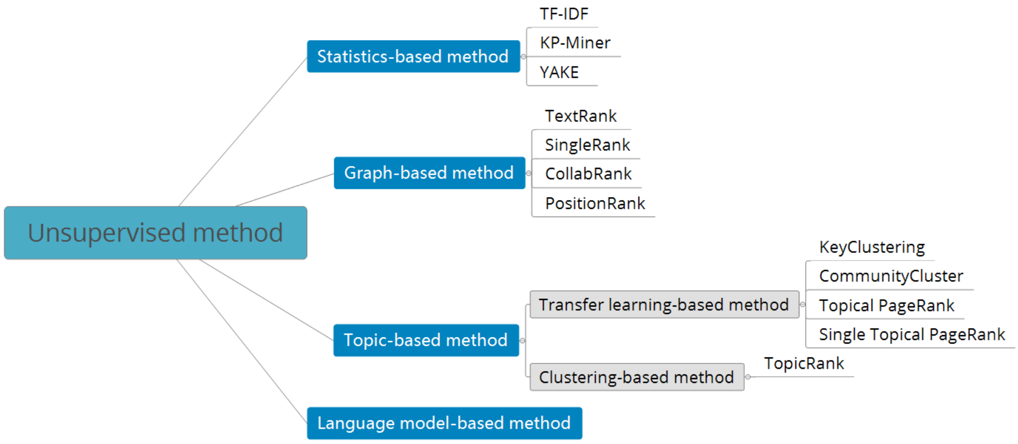


Figure 1. Summary of unsupervised methods.

The symbols in the unsupervised keyphrase extraction method in this paper are described in Table 3.

Table 3. Symbols used in this paper.

S (W/V)	The score for a word W or node V
WE (Vk, Vm)	Edge weight of node Vk and node Vm
α	damping factor
NB (V)	the neighboring node of node V

3.2. Statistics-Based Methods

TF-IDF:

The TF-IDF is a common baseline method in the keyphrase extraction field, in which the Term Frequency (TF) represents the frequency of a word in a document. To prevent the frequency of words in a long document from being too high, the TF usually uses the document length to normalize the value, that is, $TF = \frac{TN}{DL}$, in which TN represents how many times the word T appears in a specific document D, and DL represents the length of document D. The Inverse Document Frequency (IDF) represents how many documents the word T has appeared in. The main idea of the TF-IDF is that, when the frequency of T in a document is very high (that is, TF is very large) while other documents containing T are very few (that is, IDF is huge), it indicates that T has a good ability to distinguish keyphrases. Among them, $IDF = \log(\frac{DN}{DC+1})$, where DN represents the total number of documents and DC represents the number of documents containing the word T.

KP-miner:

The TF-IDF is generally only used as a statistical method applied by other unsupervised keyphrase extraction methods to calculate keyphrases' importance. For example, El-Beltagy and Rafea proposed the KP-miner [27] in 2009. This method is a typical unsupervised keyphrase extraction method using the TF-IDF, divided into three steps. The first step is to select the candidate words from documents, the second step is to calculate the candidate words' score, and the third step is to select the candidate word with the highest score as the final keyphrase. KP-miner introduced two new statistical features in the candidate word selection stage. (i) The least allowable seen frequency (lasf) factor means that only words that appear more than n times in a document can be regarded as candidate words. (ii) CutOff is based on the fact that, if a word appears after a given threshold position in a long document, it will not be a keyphrase, which means the word appearing after CutOff will be filtered out. Finally, the final keyphrases are selected by combining the candidate words' positions and the TF-IDF score. Experiments show that the efficiency of the algorithm is higher than Extractor [51] and KEA.

YAKE:

Campos et al. proposed YAKE [28] in 2018, as a typical unsupervised keyphrase extraction method using the TF-IDF. The difference between YAKE and KP-miner is that it uses candidate word locations or TF-IDF information and introduces a new feature set, which contains five features. The Word Casing (WC) reflects the cases of the candidate words. The Word Position (WP) reflects the position of a word, which means the more often the word is in the front of the document, the greater its value. The Word Frequency (WF) reflects that the higher is the frequency of a word in a document, the greater is its value. The Word Relatedness to Context (WRC) indicates the number of different words appearing on both sides of a candidate word. The Word DifSentence (WD) indicates the frequency of a candidate word in different sentences. The five values are combined to calculate $S(w)$, as shown in the following formula.

$$S(w) = \frac{WR*WP}{WC+\frac{WF}{WRC}+\frac{WD}{WR}} \tag{8}$$

Finally, the final $S(kw)$ of each candidate word is calculated by using the 3-gram model, as shown in Formula (7).

$$S(kw) = \frac{\prod_{w \in kw} S(w)}{TF(kw) * (1 + \sum\limits_{w \in kw} S(w))} \tag{9}$$

where kw represents the candidate word and TF represents the frequency of the keyphrase. The smaller is $S(kw)$, the more likely kw is to be a keyphrase.

3.3. Graph-Based Methods

The keyphrase extraction task is transformed into a graph sorting problem using a graph-based algorithm based on the basic assumption that more connections mean more important candidate words. The idea originated from PageRank [52] of Google with a basic idea of voting or recommendations, which means the graph's edges are considered votes. The more votes a node gets, the higher its score, and the more critical it is. Specifically, PageRank generates a directed graph containing all pages with a single page as a node. If there is a link pointing to B in web page A, node A in the graph has an edge pointing to B, regarded as A "voting" for B. The more votes a node receives the higher its score is and the higher its web page ranking. Moreover, the voting of high score nodes will contribute higher scores to the voted nodes [53]. Combining PageRank and word embedding [54], the performance on Chinese and English datasets exceeds TF-IDF and PositionRank.

TextRank:

Based on the idea of PageRank, Mihalcea and Tarau proposed TextRank [29] in 2004, which is the first algorithm to use PageRank for keyphrase extraction. The first thing TextRank does for a document is to delete the function words in the document. Only certain words with fixed parts of speech (such as adjectives and names) can be candidate words. The algorithm then links the selected candidate words according to the co-occurrence relationships between words to generate a directed and powerless graph. The initial score of each node is 1. If two words are within the window of w (w takes a random value from 2 to 20), the two words are connected by lines in the graph. Next, PageRank is run to calculate each node's final score, where the score of node VK is determined by the Formula (8). Finally, the document's consecutive candidate words will be connected into multi-word keyphrases, where the score is the sum of the scores of each candidate word, and the top-ranked candidate words are taken as the keyphrases.

$$S(Vk) = (1 - \alpha) + \alpha \sum_{m \in NB(Vk)} \frac{1}{|NB(Vm)|} S(Vm) \tag{10}$$

To prevent TextRank from encountering a dead cycle in the recursive computations, it introduces a damping factor (α). NB (vi) represents the neighboring node set of node vi.

SingleRank:

In view of the fact that the graphs constructed by TextRank are unweighted graphs and the weights of the edges can reflect the strength of the semantic relationship between the two nodes, using the weighted graph may be better in the keyphrase extraction task. Based on this assumption, Wan and Xiao proposed SingleRank [30] in 2008, which added weights on the basis of the TextRank between nodes appearing in the window of w at the same time, and the weight value was determined by the number of times the two words appeared in the window of w at the same time. The final score of nodes is determined by Formula (9), where $C(Vj, Vm)$ represents the number of times that nodeVj and node Vm appear together in a document.

$$S(Vk) = (1 - \alpha) + \alpha \sum_{m \in NB(Vk)} \frac{C(Vj, Vm)}{\sum\limits_{Vk \in NB(Vm)} C(Vm, Vj)} S(Vm) \tag{11}$$

ExpandRank:

In 2008, based on SingleRank, Wan and Xiao proposed ExpandRank [30], which takes the neighboring documents in the same dataset into account to provide the background knowledge when extracting keyphrases from a specific document. Specifically, ExpandRank first uses vectors to represent the documents in the dataset. Next, it calculates the k neighboring documents similar to the extracted document d0 to form a k + 1 document set D. Then, it builds a global graph to assist in extracting the keyphrases by using D, where the edge weight WE (Vk, Vm) between nodes Vk and Vm in the global graph is determined by Formula (10). Sim (d0,di) represents the similarity of documents d0 and di and Fdi (Vk,Vm) represents the number of times that nodes Vk and Vm appear in document di at the same time. The efficiency of ExpandRank is not significantly better than that of SingleRank.

$$WE(Vk, Vm) = \sum_{di \in D} sim(d0, di) \cdot Fdi(Vk, Vm) \tag{12}$$

PositionRank:

Florescu et al. proposed PositionRank [55] in 2017, which introduces location information based on SingleRank according to the idea that the earlier the candidate words appear in a document, the more important they are. As shown in Formula (11), where each item in vector P represents the normalized location information of a candidate word, the final score of each candidate word can be calculated by bringing the location information of each node into Formulas (12) and (13), where pk is the kth element in P, that is, the ratio of the position of the kth candidate word to the sum of positions of all candidate words; w is the weight of the edge; and $adj(v)$ is the adjacent node of v.

$$P = [\frac{p1}{p1 + p2 + \ldots + pn}, \frac{p2}{p1 + p2 + \ldots + pn}, \ldots, \frac{pn}{p1 + p2 + \ldots + pn}] \tag{13}$$

$$S(Vk) = (1 - \alpha)pk + a \cdot \sum_{vm \in adj(Vk)} \frac{Wmk}{O(Vm)} S(Vm) \tag{14}$$

$$O(Vm) = \sum_{vi \in adj(Vm)} Wmi \tag{15}$$

Graph-based algorithms have some disadvantages. As far as multi-topics documents (such as news) are concerned, human language habits determine that a new topic will have corresponding new keyphrases. However, in graph-based methods, all candidate words (node) are uniformly sorted, and the node with the highest score is taken as the keyphrase. This does not completely guarantee that the keyphrases output by the algorithm can cover all topics, and it may cause the phenomenon that all the keyphrases describe the same topic [24], which is improved by topic-based methods.

3.4. Topic-Based Methods

Topic-based methods can be further divided into transfer learning-based methods and clustering-based methods.

3.4.1. Transfer Learning-Based Methods

Applying the knowledge acquired from one problem to another different but related problem is the primary motivation of transfer learning [56]. Common knowledge in keyphrase extraction includes Wikipedia [33] and citation networks [37]. Because some background knowledge is needed to classify candidate words in topic-based methods, transfer learning is widely used. The following introduces several mainstream transfer learning-based methods.

KeyCluster:

Applying the knowledge acquired from one problem to another different but related problem is the primary motivation of transfer learning [56]. Common knowledge in keyphrase extraction includes Wikipedia [33] and citation networks [37]. In 2009, Liu et al. proposed KeyCluster [25], divided into four steps. As with other methods, the first step is to preprocess the document, delete the function words, and use the remaining words as candidate words. The second step is to use the Wikipedia-based method to calculate the semantic relationships of candidate words. The Wikipedia-based method regards each word as a vector with each item being the TF-IDF value in Wikipedia. The correlation between the two words can be measured by comparing the vector representations of the two words. The third step is to group the candidate words based on these semantic relationships and find each group's exemplar. The fourth step is to extract the final keyphrases from the exemplar. The experimental results show that the performance of KeyCluster is better than TextRank, and the extracted keyphrases cover the whole document.

CommunityCluster:

In 2009, Grineva et al. proposed CommunityCluster [26] based on the assumption that the words related to the same topic are generally aggregated into a subgraph (or community), and the most connected subgraph generally corresponds to a theme of a document. CommunityCluster uses Girvan–Newman network analysis to detect communities and uses all words in the most closely connected communities as keyphrases. According to the experimental results, CommunityCluster is superior to the baseline system, such as TF-IDF, Yahoo!, and Wikify! [57], in precision and recall.

Topical PageRank (TPR):

In 2010, Liu et al. proposed TPR [24], which uses Wikipedia articles as resources to train the potential Dirichlet Distribution (LDA) and uses the trained LDA model to calculate the topic distribution of documents. Then, it uses PageRank for each topic, as shown in Formula (15), to calculate the topic-specific importance scores. Finally, it combines these scores to calculate the candidate words' total score and selects the top-ranked word as keyphrases. TPR, similar to KeyCluster, ensures that the extracted keyphrases cover the entire document. According to the experimental results, TPR is better than the baseline methods, such as TF-IDF and PageRank, in precision, recall, F-score, Bpref, MRR, and MR.

$$St(Vk) = \alpha \sum_{m:Vm \rightarrow Vk} \frac{Wmk}{O(Vm)} St(Vm) + (1-\alpha) pt(Vk) \tag{16}$$

Here, t represents a topic and pt represents the LDA distribution of t.

It is worth mentioning that the introduction of LDA makes each topic have different weights when using TPR, and topics with low weights may not output related keyphrases, which is more in line with human language habits. For example, when we write an article on natural language processing, we may use 20% of the content to describe human language habits, 70% of the content to write about how a computer deals with human language, and 10% to write other things, and this 10% may not be needed to extract keyphrases, which is a feature that KeyCluster does not have.

Single Topical PageRank:

Because the TPR needs to run PageRank once for each topic, its running efficiency is reduced. Based on this weakness of TPR, Sterckx et al. improved TPR in 2015 and proposed Single Topical PageRank (Single TPR) [58]. Single TPR only needs to run PageRank once for a document, which significantly improves the running efficiency on the premise of accuracy, especially when dealing with large datasets.

The topic-based method includes the use of transfer learning and the use of hierarchical aggregative clustering to complete the keyphrase extraction task.

3.4.2. Clustering-Based Methods

TopicRank:

In 2013, Bougouin et al. proposed topicRank [22] that is similar to TextRank using candidate words as graph nodes, as TopicRank uses topics as graph nodes. Specifically, TopicRank first uses hierarchical agglomerative clustering [33] to divide the document into multiple topics, uses PageRank to score each topic, then selects the first candidate word from each top-ranked topic, and finally uses all the selected candidate words as the keyphrases. According to the experimental results, the method makes the extracted keyphrases cover all topics, and the performance is better than TF-IDF, SingleRank, and TextRank in precision, recall, and F-score.

3.5. Language Model-Based Methods

Based on Kullback–Leibler (KL) divergence that can measure the loss of two language models, Tomokiya et al. used two kinds of datasets with different functions, foreground corpus and background corpus, to assist in keyphrase extraction [59]. The foreground corpus is the dataset for keyphrase extraction, while the background corpus provides background knowledge. Similar to TF-IDF, this method reflects each keyphrase's unique extent by using background knowledge and introduces two new features, namely phraseness and informativeness. Phraseness represents the extent to which a word sequence can be used as a phrase, while informativeness represents the extent to which the phrase can express a document's central idea. This method uses the n-gram model to learn these two features in the foreground corpus and the background corpus. The phraseness and informativeness determine the final scores of the candidate words.

In the above three types of methods (statistics-based methods, graph-based methods, and topic-based methods), each method often contains more than one idea. For example, TPR uses two ideas of topic and graph. This connection is described in detail in Figure 2.

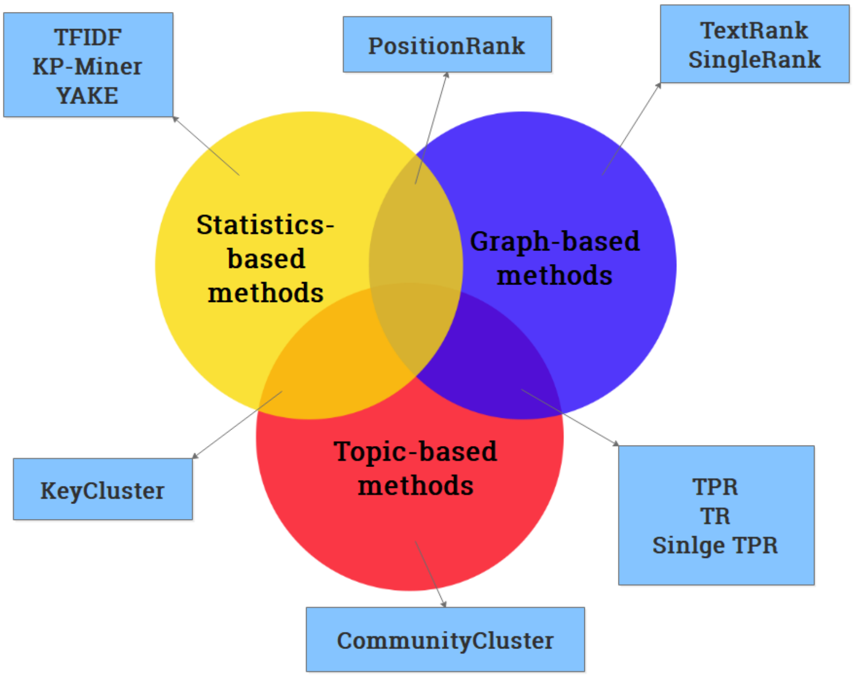


Figure 2. All methods are classified according to the technology applied. The overlapping part represents that the method uses multiple technologies.

4. The State of the Art

The performances of the mainstream unsupervised methods in the keyphrase extraction field are tested and analyzed in this section, including the statistics-based method TF-IDF; the graph-based methods TextRank, SingleRank, and PositionRank; and the topic-based methods TPR and TR. Each algorithm outputs ten keyphrases. The experimental datasets selected are mainly from the abstracts of academic papers (KDD, WWW, and Nguyen) and news (DUC-2001), which are benchmark datasets in this field, as detailed in Section 2.1. The experimental results are shown in Table 4.

Table 4. Scores achieved on various datasets (P, R, and F1 are the abbreviations of precision, recall, and F1-score, respectively).

Dataset	Method	P%	R%	F1%	Dataset	Method	P%	R%	F1%
DUC	TF-IDF	9.4	12.4	10.6	KDD	TF-IDF	9.2	23.3	12.9
	TextRank	11.1	14.1	12.2		TextRank	6.1	13.6	7.9
	SingleRank	21.5	27.4	23.8		SingleRank	7.1	16.1	9.2
	PositionRank	18.9	24.8	21.2		**PositionRank**	**9.6**	**24.3**	**13.4**
	TR	18.2	23.3	20.2		TR	6.3	14.1	8.2
	TPR	**22.3**	**28.2**	**24.6**		TPR	7.2	16.5	9.5
Nguyen	TF-IDF	10.6	26.2	14.8	WWW	TF-IDF	9.7	21.6	13.2
	TextRank	6.7	16.6	9.3		TextRank	6.6	13.3	8.1
	SingleRank	8.4	21.2	11.8		SingleRank	7.6	15.4	9.5
	PositionRank	**11.3**	**28.1**	**15.7**		**PositionRank**	**11.1**	**23.7**	**14.4**
	TR	8.1	19.6	11.1		TR	7.3	14.1	8.9
	TPR	8.7	21.8	12.1		TPR	7.9	16.1	9.9

In terms of the abstracts of academic papers (KDD, WWW, and Nguyen), it can be found using the precision, recall, and F1-score that PositionRank has the best performance, followed by TF-IDF, TPR, SingleRank, TR, and TextRank, among which TPR and SingleRank are almost the same, as shown in Figure 3.

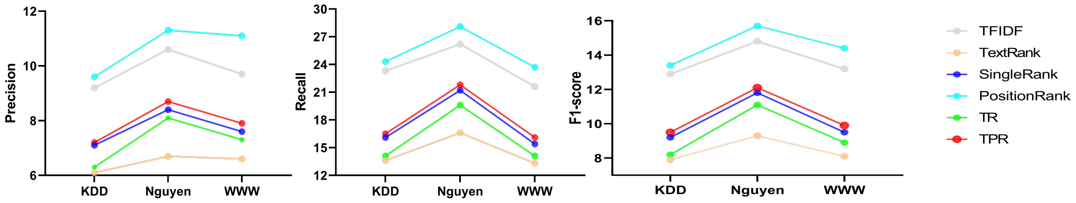


Figure 3. The performance of state of the art evaluated on KDD, Nguyen, and WWW, which are all paper abstract datasets (single topic). Overall, the performance of PositionRank and TF-IDF is higher than other methods (see Section 5.1 for more details). Precision, recall, and F1-score are introduced in Section 2.2.

5. Analysis

Although some researchers have introduced the performance differences of each method in their work, as far as we know, they have not pointed out why [26]; thus, in this section, we analyze the performance of each keyphrase extraction method from two perspectives. First, from each method's characteristics, we analyze why their performance is different (Section 5.1). Secondly, we start from the characteristics of the datasets themselves and show how different datasets affect the performance of the methods (Section 5.2).

5.1. The Performance of the Methods

Overall, the performance of PositionRank and TF-IDF is higher than other methods. We infer that, compared with other methods, these two methods use statistical data to reflect the document's important content better. The performance of the PositionRank is higher than that of TF-IDF because PositionRank not only uses statistical data but also uses a graph method, which further reflects the structure of the document.

In the graph-based method, SingleRank adds the weight of the edge in constructing the graph based on TextRank, and the weight of the edge can reflect the strength of the semantic relationship between the two words so that it can increase the performance. Based on SingleRank, PositionRank takes the position information of words into account, which is in line with human language habits that important words often appear in the front of the article, which improves the algorithm's accuracy.

In the topic-based methods, TR uses the given document itself (single document) for keyphrase extraction, while TPR trains the LDA model using Wikipedia to learn the relevant knowledge and then uses the trained model to extract keyphrases. It is evident that Wikipedia resources are much greater than a single document, so TPR has better performance than TP.

5.2. The Impacts of Dataset on Performance

While using the news dataset (DUC-2001) to test each method's performance, it is found that the methods show different performance compared with the paper's abstract dataset, as shown in Figure 4.

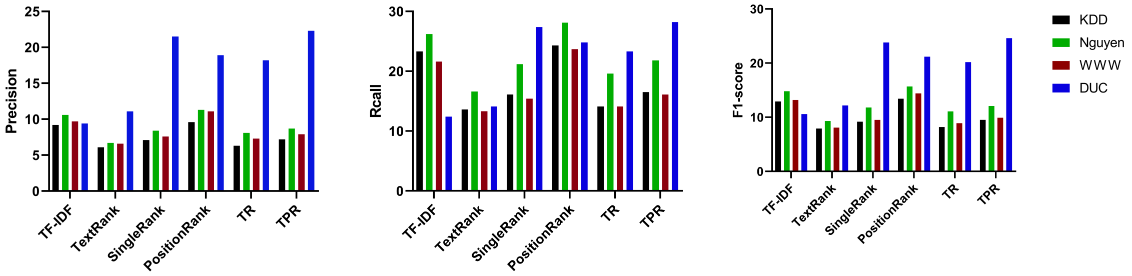


Figure 4. The performance of state of the art evaluated on KDD, Nguyen, and WWW, which are all paper abstract datasets (single topic), and DUC-2001, which is a news dataset (multiple topics). The topic-based method (TR,TPR) greatly improves the performance on multi-topic datasets (see Section 5.2 for more details). Precision, recall and F1-score are introduced in Section 2.2.

Th precision and F1-score change slightly for the TF-IDF method, but the recall is significantly reduced. For example, the precision, recall, and F1-score of TF-IDF on the KDD, Nguyen, and WWW datasets are 9.2–10.6% (9.8% on average), 21.6–26.2% (23.7% on average), and 12.9–14.8% (13.6% on average), respectively. The precision, recall, and F1-score on DUC-2001 are 9.4%, 12.4%, and 10.6%, respectively, which is decreased by 4%, 40%, and 22%, respectively. Similarly, in the graph-based methods, TextRank and PositionRank have significantly higher precisions and F1-scores, but there is little change in the recall. Specifically, the two methods, respectively, experienced 71% and 77% higher precisions, 45% and 46% higher F1-scores, and 3% and 2% lower recall. SingleRank significantly improved in all evaluation indexes, including its precision, recall, and F1-score increasing by 180%, 56%, and 133%, respectively. In the topic-based methods, th e TP and TPR have significantly improved in all evaluation indexes. Specifically, TR had a 152%, 47%, and 115% higher precision, recall, and F1-score, respectively, and TPR had a 182%, 56%, and 146% higher precision, recall, and F1-score, respectively.

The following conclusions can be drawn. Because news articles usually contain more topics than paper abstracts and the recall can reflect the comprehensiveness of a method, as described in Section 2.2, the recall of TF-IDF on multi-topic news datasets is reduced, which shows that this method cannot effectively cover all topics of an article, and the extracted keyphrases are not comprehensive enough. Further analysis shows that this is because the TF-IDF holds that, if a keyphrase appears more often in a document and less frequently in other documents, it will get a higher score. However, the numbers of keyphrases corresponding to each topic in multi-topic documents are often different (related to the topic); therefore, the candidate words in important topics may cover all the final keyphrases, while the keyphrases corresponding to unimportant topics are ignored. As a result, this method cannot guarantee the comprehensiveness of the keyphrase extraction. Based on this, it can be concluded that the TF-IDF is more suitable for the keyphrase extraction of single topic documents.

In the graph-based methods, TextRank calculates each candidate word's score using PageRank, which reflects the semantic relationships between words. Therefore, the precision and F1-score are improved for the multi-topic datasets. However, similar to the TF-IDF, TextRank cannot cover all topics well and guarantee the comprehensiveness of the extracted keyphrases, and so the recall does not change much. For SingleRank, it adds weight to the edges based on TextRank, strengthening the semantic connection between words. Therefore, the precision, F1-score, and recall have all been greatly improved. PositionRank considers the first occurrence of words based on SingleRank, but it has lower performance because the essential words in the abstract will appear in front of the article, but the news does not have this feature. It can be concluded that TextRank and PositionRank are more suitable for the keyphrase extraction of single topic documents, while SingleRank can be used for single documents and multi-topic documents.

The topic-based methods (TR and TPR) have a considerable advantage in the news datasets because the main idea of the TPR is to divide a document into several topics through LDA, calculate the score of each keyphrase corresponding to the related topics, and select the final keyphrase based on these scores. Moreover, TR divides the document into multiple topics using hierarchical agglomerative clustering, uses PageRank to score each topic, and finally selects the top topics' final keyphrases. TPR and TR have well reflected the semantic associations between topics in documents; therefore, their performances have been greatly improved, and there is no doubt that these two methods are more suitable for the keyphrase extraction of multi-topic documents.

6. Limitation of Keyphrase Extraction Methods

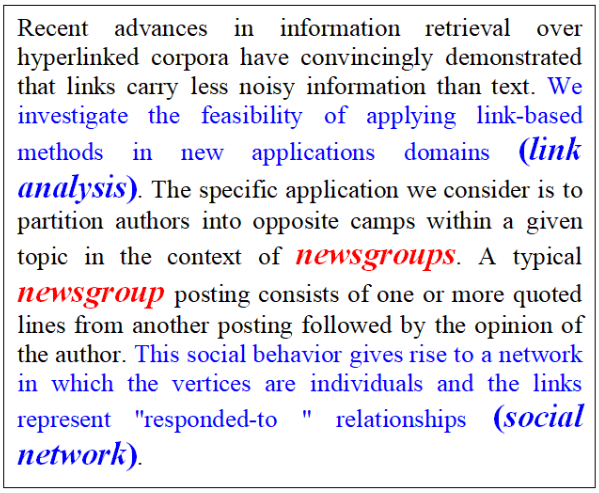
The limitation of the keyphrase extraction task makes various unsupervised methods unable to complete the task well. One is the impact of the "gold standard" problem on evaluation (Section 6.1), and the second is due to the habit of artificially annotating the datasets, resulting in the algorithm being unable to extract keyphrase that is consistent with the artificial annotation label (Section 6.2). Based on these two limitations, we discuss the possible ways to solve these problems and provide some new features that may improve the keyphrase extraction method's performance in Section 6.3.

6.1. The Impact of Gold Standard on Evaluation

The precision, recall, and F-score have a common disadvantage that each extracted keyphrase is considered correct only when it is entirely consistent with the gold standard keyphrase, which will cause two problems. The first is that the extracted keyphrase has the same root but different expression forms with the gold standard keyphrase. For example, if the gold standard keyphrase is "concept of wealth", the algorithm will not use the "conception of wealth" as a keyphrase, which is not what we want. This phenomenon is called the exact match problem, which can be solved using a stemmer. The keyphrases can be reduced to their root form and then compared with the stemmer. The second is that the extracted keyphrase and gold standard keyphrase have the same semantics but different words. For example, the gold standard keyphrase is "data processing", while the algorithm will not use "data handling" as the keyphrase. At present, there is no right solution.

6.2. The Impact of Manually Assigned Labels on Evaluation

The current keyphrase extraction method takes the words or phrases in an article as the final keyphrases. However, some keyphrases that are manually assigned are not the words appearing in the original document (it may be a summary of the semantics of content in the original document), and the keyphrase extraction method cannot extract such keyphrases, which is the limitation of the method. On some datasets, the error caused by this problem can reach 52–73%. For example, Figure 5 is a document from WWW, which contains five keyphrases: link analysis, newsgroup, social network, text mining, and web mining. Among them, only "newsgroup" has appeared in the original document, while the other keyphrases are summary expressions of the meaning of the content. The blue sentence in the figure is the sentence expressing the keyphrases, the blue italics and bold words are the keyphrases for which their original words did not appear in the document, and the red italics and bold words are the keyphrases for which original words appear in the document.



Recent advances in information retrieval over hyperlinked corpora have convincingly demonstrated that links carry less noisy information than text. We investigate the feasibility of applying link-based methods in new applications domains ***(link analysis)***. The specific application we consider is to partition authors into opposite camps within a given topic in the context of ***newsgroups***. A typical ***newsgroup*** posting consists of one or more quoted lines from another posting followed by the opinion of the author. This social behavior gives rise to a network in which the vertices are individuals and the links represent "responded-to " relationships ***(social network)***.

Figure 5. The impact of manually assigned labels on evaluation. Only "newsgroups" can be extracted by the algorithm.

6.3. Our Recommendations

For the impact of the gold standard on evaluation, we need to introduce external knowledge to determine whether the keyphrases extracted by a method and the gold keyphrase have the same semantics. For example, we can use external resources to train Word2vec [54], a commonly used model in the NLP field, to assist in the task of keyphrase extraction. Specifically, we can use the trained Word2vec model to convert each extracted keyphrase into an embedding vector, and then compare the similarity with the gold keyphrase. If the similarity is higher than a certain threshold, the extraction can be considered successful.

Regarding the influence of manually assigned labels on evaluation, on the one hand, through observation of the datasets, we find that the original words can directly express many keyphrases in the document without artificial summary. Thus, we can start from this aspect when constructing the dataset and select the words that have appeared in the article as the gold keyphrase. On the other hand, similar to solving the gold standard problem, we can also introduce external knowledge into the method. For example, we can use Word2vec to understand the meaning of the sentence by constructing the embedding vector and then find the words that are most relevant to the meaning of the sentence as keyphrases.

For these recommendations, we have done some experiments for future researchers to study further. We use Word2vec of gensim (a Python library) to try to solve the "gold standard" and "manually assigned labels" problems.

We use PositionRank to test on the WWW dataset. Unlike the usual evaluation method, the extracted keyphrases and gold keyphrases will not be restored to the root form with a stemmer, nor will they be directly matched with strings. We use the trained Word2vec model to convert the extracted keyphrase and gold keyphrase into vectors, and then calculate the Euclidean distance or cosine similarity of the two vectors. If the Euclidean distance is less than a certain threshold or the cosine similarity is higher than a certain threshold, we manually compare whether the words corresponding to the two vectors have similar semantics. If so, the extracted keyphrases and gold keyphrases are considered to match.

The experimental results show that using Word2vec to evaluate the performance of the method can better reflect the similarity of extracted keyphrases than the conventional evaluation metrics (precision, recall, and F1-score), and it can also help extraction method to extract keyphrases with different from but similar semantics with gold keyphrases.

However, we also encounter some difficulties in setting a threshold (Euclidean distance or cosine similarity) to determine whether an extracted keyphrase can be considered as a gold keyphrase because the threshold is an empirical parameter that requires much labor to compare the semantic similarity between the extracted keyphrases and the gold keyphrases. We will continue to study this idea further.

7. Conclusions and Future Directions

In this paper, the unsupervised learning methods in the field of keyphrase extraction are summarized, and the performance of each method on different datasets and the reasons for the performance difference are analyzed in detail, to help future researchers understand mainstream solutions in the field of keyphrase extraction from multiple perspectives.

The reasons for the limitations of the keyphrase extraction field are pointed (“gold standard” and ”manually assigned labels”), and our recommendations to solve these problems are proposed.

To help researchers further improve the performance of the method in the keyphrase extraction task, we introduce some new features that may be helpful.

The relative position of words: It is found that many methods, such as PositionRank and YAKE, have introduced the position information of the word, that is, the positions of the word in the full document. However, for long texts with multiple topics, keyphrase will appear with the appearance of new topics, so some keyphrase will appear in topics that are located later, that is, the keyphrase will be relatively far away from the beginning of the article and the previous location-based features would not give these words high weight. Based on this discovery, instead of using an entire document as a reference, we can use each paragraph as an independent unit to calculate the position of the word from the beginning of the paragraph.

The role of conjunctions: It is worth mentioning that, if the second clause of two comma-connected clauses starts with a conjunction, there is likely to be some semantic relationship between the two sentences [60]. Based on this discovery, paying attention to the function and position of conjunctions in the keyphrase extraction process may help to improve the performance.

Design method for different types of datasets: In Section 5.2, we show how the datasets affect the performance of methods; thus, in future work, researchers should design different methods based on the characteristics of the datasets.

Author Contributions: Conceptualization, L.C. and C.S.; methodology, C.S.; software, C.S.; validation, C.S., L.C.; formal analysis, C.S.; investigation, T.L.; resources, T.L.; data curation, T.L.; writing—original draft preparation, C.S.; writing—review and editing, S.L.; visualization, S.L.; supervision, L.C.; project administration, H.L.; funding acquisition, L.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key R&D Plan of China under Grant No. 2017YFA0604500, and by National Sci-Tech Support Plan of China under Grant No. 2014BAH02F00, and by National Natural Science Foundation of China under Grant No. 61701190, and by Youth Science Foundation of Jilin Province of China under Grant No. 20160520011JH & 20180520021JH, and by Youth Sci-Tech Innovation Leader and Team Project of Jilin Province of China under Grant No. 20170519017JH, and by Key Technology Innovation Cooperation Project of Government and University for the whole Industry Demonstration under Grant No. SXGJSF2017-4, and by Key scientific and technological R&D Plan of Jilin Province of China under Grant No. 20180201103GX and by Project of Jilin Province Development and Reform Commission under Grant No. 2019FGWTZC001.

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

References

1. Babar, S.A.; Patil, P.D. Improving Performance of Text Summarization. *Procedia Comput. Sci.* **2015**, *46*, 354–363. [CrossRef]
2. Welleck, S.; Brantley, K.; Daumé, H., III; Cho, K. Non-Monotonic Sequential Text Generation. *arXiv* **2019**, arXiv:1902.02192.
3. Welleck, S.; Kulikov, I.; Roller, S.; Dinan, E.; Cho, K.; Weston, J. Neural Text Generation with Unlikelihood Training. *arXiv* **2019**, arXiv:1908.04319.
4. Puduppully, R.; Dong, L.; Lapata, M. Data-to-Text Generation with Content Selection and Planning. *arXiv* **2019**, arXiv:1809.00582.
5. Shen, S.; Fried, D.; Andreas, J.; Klein, D. Pragmatically Informative Text Generation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4060–4067.
6. Mallett, D.; Elding, J.; Nascimento, M.A. Information-content based sentence extraction for text summarization. In Proceedings of the International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA, 5–7 April 2004; IEEE: Las Vegas, NV, USA, 2004; Volume 2, pp. 214–218 .
7. Bougouin, A.; Boudin, F.; Daille, B. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. In Proceedings of the IJCNLP, Nagoya, Japan, 14–18 October 2013.
8. Liu, Z.; Liang, C.; Sun, M. Topical Word Trigger Model for Keyphrase Extraction. Available online: https://www.aclweb.org/anthology/C12-1105.pdf (accessed on 12 June 2020).
9. Azad, H.K.; Deepak, A. Query expansion techniques for information retrieval: A survey. *Inf. Process. Manag.* **2019**, *56*, 1698–1735. [CrossRef]
10. Guo, J. A Deep Look into Neural Ranking Models for Information Retrieval. *arXiv* **2019**, arXiv:1903.06902.
11. Gutierrez, C.E.; Alsharif, M.R. A Tweets Mining Approach to Detection of Critical Events Characteristics using Random Forest. *Int. J. Next Gener. Comput.* **2014**, *5*, 167–176.
12. Gutierrez, C.E.; Alsharif, M.R.; He, C.; Khosravy, M.; Villa, R.; Yamashita, K.; Miyagi, H. Uncover news dynamic by principal component analysis. *ICIC Express Lett.* **2016**, *7*, 1245–1250.
13. Dave, K.; Lawrence, S.; Pennock, D. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proceedings of the WWW'03, Budapest, Hungary, 20–24 May 2003.
14. Hemmatian, F.; Sohrabi, M.K. A survey on classification techniques for opinion mining and sentiment analysis. *Artif. Intell. Rev.* **2019**, *52*, 1495–1545. [CrossRef]
15. Asghar, M.Z.; Khan, A.; Zahra, S.R.; Ahmad, S.; Kundi, F.M. Aspect-based opinion mining framework using heuristic patterns. *Cluster. Comput.* **2019**, *22*, 7181–7199. [CrossRef]
16. Frank, E.; Paynter, G.W.; Witten, I.; Gutwin, C.; Nevill-Manning, C. Domain-Specific Keyphrase Extraction. In Proceedings of the IJCAI, Stockholm, Sweden, 31 July–6 August 1999.
17. Hulth, A.; Megyesi, B.B. A Study on Automatically Extracted Keywords in Text Categorization. Available online: https://www.aclweb.org/anthology/P06-1068.pdf (accessed on 15 June 2020).
18. Turney, P.D. *Learning to Extract Keyphrases from Text*; Technical Report; National Research Council, Institute for Information Technolog: Ottawa, ON, Canada, 2002.
19. Witten, I.H.; Paynter, G.W.; Frank, E.; Gutwin, C.; Nevill-Manning, C.G. KEA: Practical Automatic Keyphrase Extraction. In Proceedings of the Fourth ACM Conference on Digital Libraries, Berkeley, CA, USA, 11–14 August 1999.
20. Wang, R.; Wang, G. Web Text Categorization Based on Statistical Merging Algorithm in Big Data. *Environ. Int. J. Ambient. Comput. Intell.* **2019**, *10*, 17–32. [CrossRef]
21. Gutierrez, C.E.; Alsharif, M.R.; Khosravy, M.; Yamashita, K.; Miyagi, H.; Villa, R. Main Large Data Set Features Detection by a Linear Predictor Model. Available online: https://aip.scitation.org/doi/abs/10.1063/1.4897836 (accessed on 20 June 2020).
22. Salton, G.; Buckley, C. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.* **1988**, *24*, 513–523. [CrossRef]
23. Kim, S.N.; Kan, M.Y. Re-examining automatic keyphrase extraction approaches in scientific articles. In Proceedings of the Workshop on Multiword Expressions Identification, Interpretation, Disambiguation and Applications—MWE'09, Singapore, 25–27 November 2009; p. 9.

24. Liu, Z.; Huang, W.; Heng, Y.; Sun, M. *Automatic Keyphrase Extraction via Topic Decomposition. Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Singapore, 2010.
25. Liu, Z.; Li, P.; Zheng, Y.; Sun, M. Clustering to find exemplar terms for keyphrase extraction. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing—EMNLP'09, Singapore, 6–7 August 2009; Association for Computational Linguistics: Singapore, 2009; Volume 1, p. 257.
26. Hasan, K.S.; Ng, V. Automatic Keyphrase Extraction: A Survey of the State of the Art. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 23–25 June 2014; Association for Computational Linguistics: Baltimore, MD, USA, 2014; pp. 1262–1273.
27. El-Beltagy, S.R.; Rafea, A. KP-Miner: A keyphrase extraction system for English and Arabic documents. *Inf. Syst.* **2009**, *34*, 132–144. [CrossRef]
28. Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.M.; Nunes, C.; Jatowt, A. YAKE! Collection-Independent Automatic Keyword Extractor. In *Advances in Information Retrieval*; Pasi, G., Piwowarski, B., Azzopardi, L., Hanbury, A., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 10772, pp. 806–810, ISBN 978-3-319-76940-0.
29. Mihalcea, R.; Tarau, P. TextRank: Bringing Order into Texts. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain, 25–26 July 2004.
30. Wan, X.; Xiao, J. CollabRank: Towards a collaborative approach to single-document keyphrase extraction. In Proceedings of the 22nd International Conference on Computational Linguistics—COLING'08, Manchester, UK, 18–22 August 2008; Association for Computational Linguistics: Manchester, UK, 2008; Volume 1, pp. 969–976.
31. Papagiannopoulou, E.; Tsoumakas, G. A review of keyphrase extraction. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2019**, e1339. [CrossRef]
32. Krapivin, M.; Autayeu, A.; Marchese, M. *Large Dataset for Keyphrases Extraction*; Technical Report DISI-09-055; DISI, University of Trento: Trento, Italy, 2009.
33. 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—EMNLP'09, Singapore, 6–7 August 2009; Association for Computational Linguistics: Singapore, 2009; Volume 3, p. 1318.
34. Nguyen, T.D.; Kan, M.-Y. Keyphrase Extraction in Scientific Publications. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*; Goh, D.H.-L., Cao, T.H., Sølvberg, I.T., Rasmussen, E., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4822, pp. 317–326, ISBN 978-3-540-77093-0.
35. Schutz, A. Keyphrase Extraction from Single Documents in the Open Domain Exploiting Linguistic and Statistical Methods. Available online: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.394.5372&rep=rep1&type=pdf (accessed on 30 June 2020).
36. Hulth, A. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, Sapporo, Japan, 11–12 July 2003; Volume 10, pp. 216–223.
37. Gollapalli; Sujatha, D.; Cornelia, C. Extracting Keyphrases from Research Papers Using Citation Networks. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; pp. 1629–1635.
38. Jiang, M.; Chen, Y.; Liu, M.; Rosenbloom, S.T.; Mani, S.; Denny, J.C.; Xu, H. A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. *J. Am. Med. Inform. Assoc.* **2011**, *18*, 601–606. [CrossRef] [PubMed]
39. Bougouin, A.; Barreaux, S.; Romary, L.; Boudin, F.; Daille, B. TermITH-Eval: A French Standard-Based Resource for Keyphrase Extraction Evaluation. In Proceedings of the Language Resources and Evaluation Conference (LREC), Portorož, Slovenia, 23–28 May 2016.
40. Wan, X.; Xiao, J. Single Document Keyphrase Extraction Using Neighborhood Knowledge. Available online: https://www.aaai.org/Papers/AAAI/2008/AAAI08-136.pdf (accessed on 25 June 2020).
41. Marujo, L.; Gershman, A.; Carbonell, J.; Frederking, R.; Neto, J.P. Supervised Topical Key Phrase Extraction of News Stories using Crowdsourcing, Light Filtering and Co-reference Normalization. In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, 23–25 May 2012.

42. Marujo, L.; Viveiros, M.; Neto, J.P. Keyphrase Cloud Generation of Broadcast News. In Proceedings of the 12th Annual Conference of the International Speech Communication Association (Interspeech), Florence, Italy, 27–31 August 2011.
43. Grineva, M.; Grinev, M.; Lizorkin, D. Extracting Key Terms from Noisy and Multitheme Documents. Available online: https://dl.acm.org/doi/abs/10.1145/1526709.1526798 (accessed on 26 June 2020).
44. Hammouda, K.M.; Matute, D.N.; Kamel, M.S. CorePhrase: Keyphrase Extraction for Document Clustering. In *Machine Learning and Data Mining in Pattern Recognition*; Perner, P., Imiya, A., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3587, pp. 265–274, ISBN 978-3-540-26923-6.
45. Voorhees, E.M.The TREC-8 question answering track report. In Proceedings of the Eighth Text Retrieval Conference, TREC 1999, Gaithersburg, MD, USA, 17–19 November 1999.
46. Buckley, C.;Voorhees, E.M. Retrieval Evaluation with Incomplete Information. Available online: https://dl.acm.org/doi/abs/10.1145/1008992.1009000 (accessed on 18 June 2020).
47. Blei, D.M. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
48. Halliday, M.A.K.; Matthiessen, C.M.I.M. *An Introduction to Functional Grammar*, 3rd ed.; Distributed in the United States of America by Oxford University Press: Oxford, MS, USA, 2004; ISBN 978-0-340-76167-0.
49. Johansson, V. Lexical Diversity and Lexical Density in Speech and Writing: A Developmental Perspective. Working Papers. Available online: https://www.semanticscholar.org/paper/Lexical-diversity-and-lexical-density-in-speech-and-Johansson/f0ec9ed698d5195220f80b732e30261eafbe1ad8?p2df (accessed on 10 June 2020).
50. Yih, W.; Goodman, J.; Carvalho, V.R. Finding advertising keywords on web pages. In Proceedings of the 15th International Conference on World Wide Web—WWW'06, Edinburgh, UK, 23–26 May 2006; p. 213.
51. Turney, P.D. Learning Algorithms for Keyphrase Extraction. *Inf. Retr.* **2000**, *2*, 303–336. [CrossRef]
52. Page, L.; Brin, S.; Motwani, R.; Winograd, T. The PageRank Citation Ranking: Bringing Order to the Web. Available online: http://ilpubs.stanford.edu:8090/422/ (accessed on 5 June 2020).
53. Wang, H.; Ye, J.; Yu, Z.; Wang, J.; Mao, C. Unsupervised Keyword Extraction Methods Based on a Word Graph Network. *Int. J. Ambient. Comput. Intell.* **2020**, *11*, 68–79. [CrossRef]
54. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. Available online: https://ui.adsabs.harvard.edu/abs/2013arXiv1301.3781M/abstract (accessed on 1 June 2020).
55. Florescu, C.; Caragea, C. PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1105–1115.
56. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
57. Mihalcea, R.; Csomai, A. Wikify!: Linking documents to encyclopedic knowledge. In Proceedings of the Sixteenth ACM Conference on Conference on information and Knowledge Management—CIKM'07, Lisbon, Portugal, 6–10 November 2007; p. 233.
58. Sterckx, L.; Demeester, T.; Deleu, J.,; Develder, C. Topical word importance for fast keyphrase extraction. Presented at the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, 18–22 May 2015; pp. 121–122. [CrossRef]
59. Tomokiyo, T.; Hurst, M. A language model approach to keyphrase extraction. In Proceedings of the ACL 2003 Workshop on Multiword Expressions Analysis, Acquisition and Treatment, Sapporo, Japan, 12 July 2003; Volume 18, pp. 33–40.
60. Jernite, Y.; Bowman, S.R.; Sontag, D. Discourse-Based Objectives for Fast Unsupervised Sentence Representation Learning. *arXiv* **2017**, arXiv:1705.00557.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.