


Article

# Key Concept Identification: A Comprehensive Analysis of Frequency and Topical Graph-Based Approaches

Muhammad Aman <sup>1,\*</sup>, Abas bin Md Said <sup>1</sup>, Said Jadid Abdul Kadir <sup>1</sup> and Israr Ullah <sup>2</sup> 

<sup>1</sup> Department of Computer and Information Sciences, Universiti Teknologi Petronas, Seri Iskandar 32610, Malaysia; abass@utp.edu.my (A.b.M.S.); saidjadid.a@utp.edu.my (S.J.A.K.)

<sup>2</sup> Department of Computer Engineering, Jeju National University, Jeju 63243, Korea; israrullahkk@yahoo.com

\* Correspondence: muhammad.aman\_g03419@utp.edu.my; Tel.: +60-162-532127

Received: 9 February 2018; Accepted: 15 May 2018; Published: 18 May 2018



**Abstract:** Automatic key concept extraction from text is the main challenging task in information extraction, information retrieval and digital libraries, ontology learning, and text analysis. The statistical frequency and topical graph-based ranking are the two kinds of potentially powerful and leading unsupervised approaches in this area, devised to address the problem. To utilize the potential of these approaches and improve key concept identification, a comprehensive performance analysis of these approaches on datasets from different domains is needed. The objective of the study presented in this paper is to perform a comprehensive empirical analysis of selected frequency and topical graph-based algorithms for key concept extraction on three different datasets, to identify the major sources of error in these approaches. For experimental analysis, we have selected *TF-IDF*, *KP-Miner* and *TopicRank*. Three major sources of error, i.e., frequency errors, syntactical errors and semantical errors, and the factors that contribute to these errors are identified. Analysis of the results reveals that performance of the selected approaches is significantly degraded by these errors. These findings can help us develop an intelligent solution for key concept extraction in the future.

**Keywords:** keyphrase extraction; key concept extraction; information retrieval; empirical analysis; text mining

## 1. Introduction

The key concepts in an ontology of a specific domain represent a set of important entities' classes or objects [1,2]. Extracting these key concepts automatically is a fundamental and challenging step in Ontology Learning. In this regard, many existing approaches for extracting key concepts have focused on keyphrases extraction from text documents [1,3–7]. Keyphrases refers to terms or group of terms (phrases) within a document, that describe the document and convey its key information [1,5,8]. Because of the relatedness of both the terms *keyphrase* and *key concept*, we use them interchangeably without distinguishing between them.

Keyphrase or key concept extraction plays a basic role in many application areas. It is not limited to Ontology Learning [9] only, but also it is considered to be the core step in text and documents summarization, indexing, clustering [10], categorization [11], and currently, in improving search results [8].

While the key concepts can provide excellent means to describe a document or represent knowledge of a specific domain, the job of extracting key concepts is definitely non-trivial, as have been suggested in the recent studies [12]. Several approaches have been devised by researchers to address this problem. Broadly, these approaches can be categorized into Supervised and Unsupervised

methods. Supervised approaches for concept identification recast this task typically, as a binary classification step [13]. In these methods, a classifier is trained on annotated training documents, which classify a given phrase as *key concept* or *non-key concept* [10,14–17]). However, the effectiveness of these methods strongly relies on a large set of training documents, thus making it biased towards a specific domain and undermining their capability of generalization to other domains. A viable alternative could be an unsupervised approach.

The unsupervised methods can be categorized based on the type of techniques involved, including approaches that are based on statistical measures, i.e., TF-IDF [8,18,19], language modeling [20], graph-based weighting [21–26], and clustering techniques [27,28].

Among the above categories, statistical frequency and topical graph-based unsupervised methods are the two kinds of potentially powerful and leading approaches in this area. In order to utilize the potential of these approaches for improving key concept identification, we need to thoroughly analyze the performance of the methods based on these approaches, on datasets from different domains, and investigate the underlying reasons and error sources in case of poor results. To gain better understanding of the approaches by identifying their shortcomings, and to provide future research directions, we examine three state-of-the-art methods and evaluate their performance on three different datasets. We will describe these datasets later in the analysis section.

For our experiments, the first statistical frequency-based method we choose is TF-IDF, because it is a baseline for this approach as used in SemEval-2010 task 5 [18,19]. The second method we select is KP-Miner [8], as it is a representative method for statistical frequency-based approaches, and has outperformed all the unsupervised methods in SemEval-2010 task 5. Finally, we choose the method TopicRank [21], because it is a popular and representative method for topical clustering-based approach that has beaten the previous methods. Another reason for selecting these methods is that they are data-driven which are independent of auxiliary sources. We use the re-implementation of these methods that is publicly available [29].

This study provides a firm basis for future research work and contributes by:

- Providing a brief survey of various kinds of keyphrase extraction methods along with the necessary details and limitations of different approaches.
- Identifying the factors that can contribute to precision and recall errors in frequency and topical graph-based keyphrase extraction approaches, through performance analysis.
- Identifying the three major sources of errors in the selected approaches by conducting quantitative error source analysis.

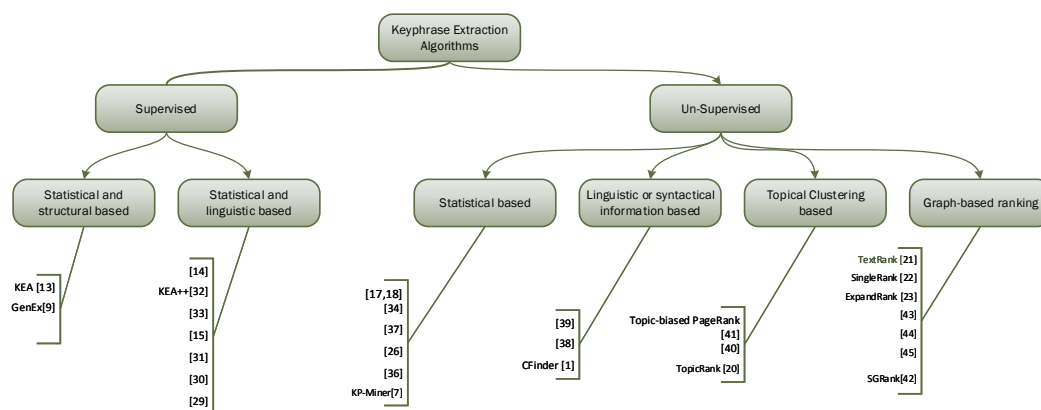
The rest of the paper is organized as follows: Section 2 presents brief survey of various supervised and un-supervised methods used for keyphrase extraction. Working of unsupervised methods is briefly explained in Section 3 with description of selected algorithms for comparative analysis. Detailed comparative analysis of selected algorithms is provided in Section 4 with error analysis. Section 5 concludes the paper with future work recommendation.

## 2. Related Work

As mentioned earlier in the introduction section, the approaches of key concept or keyphrase extraction can be broadly categorized into supervised and unsupervised methods. Various keyphrase extraction algorithms are proposed in the literature under each class. Typical classification of supervised and unsupervised methods for keyphrase extraction is given in Figure 1.

The supervised approaches reorganize this problem as binary classification issue where the main objective is to train a model on a training dataset that can classify the candidate phrases into two classes, i.e., *keyphrase* and *non keyphrase*. The classifier ranks the phrases according to their probabilities of being keyphrases. Overall performance of supervised methods is comparatively better than unsupervised methods, however, their effectiveness strongly depends on the quality of the training set that is manually annotated either by authors of the documents or domain experts. Another limitation of

supervised methods is that they are biased towards the domain, on whose training set the model is trained. The supervised methods can be further categorized based on the approach, techniques and type of features used. Initially, Frank et al. [10,14] used statistical and structural features like Term Frequency (TF) and Phrase Position. Others combined different statistical, structural and linguistic features in their algorithms [15,16,30–34].



**Figure 1.** Classification of supervised and unsupervised algorithms for keyphrase extraction.

To overcome the limitations of supervised methods, numerous unsupervised methods have been proposed that do not rely on training sets. However, the task to develop key concept extraction methods that are language independent and portable across different domains is quite challenging. The methods can be classified into several categories based on the approach and techniques used.

The methods that are based on statistical information and structural information, for example *tf-idf* (term frequency-inverse document frequency), phrase position, and topic proportion, are language independent [8,27,35–38]. However, weighting more to single terms than multiword terms and overlooking the semantics, are their main drawbacks. Despite the limitations of statistical frequency-based approach, still it is preferred approach and many algorithms are based on frequency-based model *tf-idf*, and the reason is that it is a data driven approach which is independent of auxiliary sources.

To address the issues pertaining to statistical information-based algorithms, an alternate approach is devised that exploits linguistic information and auxiliary structures. Such methods use techniques like part-of-speech tags, linguistic patterns, glossaries, WordNet, Wikipedia, or manually created semantically hierarchical databases. Auxiliary structures and linguistics based information contribute to the comprehensiveness and efficiency of keyphrase extraction, however, linguistics based techniques are language dependent and may require domain knowledge and expertise in language, while using glossaries or auxiliary structures require extensive human efforts in updating, definition of terms and terminology standardization [39,40].

CFinder [1] adopts a hybrid approach that combines techniques based on statistical, structural, linguistic-based information of candidate phrases, and domain knowledge. However, still there is a need for an optimal solution for keyconcepts identification as it is not completely domain independent.

Graph-based approach is popular among the unsupervised methods. Graph-based approaches try to overcome the limitations of aforementioned approaches by constructing a graph in which the nodes represent the candidate phrases and the edges show their relatedness. A ranking algorithm, e.g., PageRank then is used to rank the keyphrases according to their weights. Several popular graph-based systems have been proposed by researchers for example, TextRank [22], SingleRank [23], ExpandRank [24], SGRank [41]. Some other graph-based methods are recently introduced [42–45]. Most of the graph-based keyphrase extraction methods prefer single words as nodes that may result

in missing multiword phrases [1], which is one of the drawbacks of graph-based methods. Another drawback is that they do not guarantee to cover all the topics of the document [13].

Several popular topical graph-based methods exploit topic models and clustering techniques for keyphrase identification [46], Topic-biased PageRank [47], TopicRank [21]. This approach has recently attracted the attention of many researchers and is considered a potentially powerful approach.

A summary of the related work is given in Table A1 in Appendix A. It categorizes the different keyphrase extraction methods, presenting various approaches, techniques used by each method, and their limitations. Hopefully, this categorization will help in identifying future research directions.

### 3. Selected Unsupervised Methods

#### 3.1. Common Extraction Steps

Before giving a description of selected unsupervised methods for experimental analysis, we first briefly explain the working of unsupervised methods. These algorithms commonly follow three steps for a generic unsupervised keyconcept extraction.

**Candidate Phrase Selection:** In this preprocessing step, the input text is passed out through a filtration process that removes unnecessary words and produces a list of potential candidate phrases. The process is carried out using some commonly used heuristics, that include (1) filter out non-keywords using stop-word list [27], (2) considering words only with certain part-of-speech tags, i.e., nouns, adjectives [21–23,37]. Another approach is using n-grams as candidate words as reported in [8,18,19].

**Candidate Weighting:** The second step is to rank the candidate phrases. To accomplish this task, various approaches as discussed earlier, have been proposed to represent the input text, the relatedness between the candidate words, and ranking them.

**Keyconcept Formation:** The last step is to form keyconcepts from the ranked list of candidate phrases. A phrase, that is typically a sequence of nouns, verbs and adjectives is considered as keyconcept if one or more of its constituents are top ranked candidate terms [22,27], or their sum result in a top score of the phrase [23].

#### 3.2. Description of Selected Unsupervised Methods

##### 3.2.1. TF-IDF

The TF-IDF method [18,19] uses n-gram approach for candidate selection. It selects 1,2,3-grams as candidate phrases and filter out stop-words, those words consisting only punctuation marks and the words shorter than three characters. It assigns weight to each word  $w$  in a document  $d$  using the word's frequency in the document  $d$  referred to as  $tf$ -term frequency, and the  $idf$ -inverse document frequency. It can be defined as follows:

$$w_{ij} = tf_{ij} \times idf \quad (1)$$

where  $w_{ij}$  represents weight and  $tf_{ij}$  is the frequency of word  $t_j$ , in document  $D_i$ . The inverse document frequency  $idf$  is equal to  $\log_2 N/n$ , where  $N$  represents the total number of documents in the corpus and  $n$  is document frequency. The  $tf$  weighting is based on Hans Peter Luhn assumption [48]: "The weight of term that occurs in a document is simply proportional to the term frequency" whereas, the  $idf$  weighting is a statistical interpretation of specificity of a term that is described as [49]: "The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs".

##### 3.2.2. KP-Miner

KP-Miner [8] is a non-learning key concepts extraction method, meaning that it does not require any training. This system is also based on frequency-based statistical measure, i.e.,  $tf$ - $idf$ . KP-Miner emphasizes on both candidate words selection and their weighting process. Along with  $TF$  and  $IDF$ , the two other attributes used in calculating candidate's score are boosting factor and first occurrence

position. KP-Miner uses n-gram approach for candidate selection. It selects 1-5-grams as candidate phrases and filter out stop-words, those words consisting only punctuation marks and the words shorter than three characters. KP-Miner then uses two parameters to further filter out the candidate list. One is Lasf (least allowable seen frequency) that represent the minimum frequency for a candidate to be considered as key concept. Second, CuttOff constant that represents the number of words in a long document after which a candidate appears for the first time is rarely keyphrase. The values are set to 3 and 400 respectively in the original method. KP-Miner assumes that compound keyphrases are less frequent as compared to single keywords. Based on this argument, it assigns high scores to multiword keyphrases in two ways: (1) by setting document frequency to 1 for compound keyphrases, which result in maximum IDF value for such phrases, and (2) multiplying the score with a boosting factor (“related to a ratio of single to compound terms”) [8]. To calculate the weights of single or multiword candidate key concepts, the following equation is devised:

$$w_{ij} = tf_{ij} \times idf \times B_i \times P_f \quad (2)$$

where  $w_{ij}$  represents weight and  $tf_{ij}$  is the frequency of word  $t_j$ , in document  $D_i$ . The inverse document frequency  $idf$  is equal to  $\log_2 N/n$ , where  $N$  represents the total number of documents in the corpus and  $n$  is document frequency. In case of multiword candidate phrase,  $n$  is set to 1.  $P_f$  is the factor that is associated with term position. The term position  $P_f$  is set to 1, if position rules are not applied.  $B_i$  denotes the boosting factor, introduced in KP-Miner, associated with document  $D_i$ , and can be defined by the following equation:

$$B_i = \frac{|N_i|}{|P_i|^\alpha} \quad (3)$$

If  $B_i > \sigma$  then  $B_i = \sigma$ , where  $|N_i|$  represents the number of all candidate words in document  $i$ ,  $|P_i|$  is the number of all words whose length exceeds one in document  $i$ .  $\alpha$  and  $\sigma$  are weight adjustment constants. The constant  $\alpha$  controls the value of the boosting factor, without this the boosting factor would be too large, that may produce results biased towards compound words.

### 3.2.3. TopicRank

The TopicRank [21] is graph-based approach that improves SingleRank [23]. The intuition behind TopicRank is [21] “ranking topics instead of words is a more straightforward way to identify the set of keyphrases that cover the main topics of a document”. Therefore, TopicRank groups lexically similar noun phrase candidates into clusters that represent topics. Then a complete graph is constructed in which the topics are represented by vertices, and the semantic relatedness between them is denoted by edges. The weight of the edges is related to the strength of semantic relatedness between the corresponding vertices. The weight  $w$  of each edge in the graph is defined as follows:

$$w_{ij} = \sum_{c_i \in t_i} \sum_{c_j \in t_j} dist(c_i, c_j) \quad (4)$$

where,

$$dist(c_i, c_j) = \sum_{p_i \in pos(c_i)} \sum_{p_j \in pos(c_j)} \frac{1}{|p_i - p_j|} \quad (5)$$

where  $t$  represents the topic at the particular vertex of the graph  $G = (V, E)$ , and  $dist(c_i, c_j)$  is the reciprocal distances between the offset positions of the candidate concepts  $c_i$  and  $c_j$  in the given document and  $pos(c)$  refers to all the offset positions of the candidate key concept  $c$ .

The TextRank’s ranking algorithm [22] is then applied to rank the topics that ranks based on the weights of their edges. At the end, the first occurring phrase from each of the top ranked topics is extracted to form the key concepts.

#### 4. Comparative Analysis

In this section, we first describe the experimental setup, then discuss the performance of each individual method in detail, with the aim to highlight the major weaknesses of tf-idf and topical clustering-based data-driven approaches. In addition, finally, present error source analysis that provides evidence and support our arguments in performance analysis.

##### 4.1. Experimental Setup

**Data Sets:** We choose the following evaluation corpora from two different domains. (1) The SemEval-2010 task 5 benchmark data set [18,19]. This dataset includes 244 scientific articles, out of which 144 are for training and 100 are for test. (2) Quran English translation by Yousaf Ali [50,51]. (3) 500N-KPCrowd [52] dataset that is composed of news stories. The reason for selecting the datasets is that SemEval-2010 has been created in a systematic way to provide a common base for evaluation of current and future key phrase extraction systems, while the Quran translation is selected because its contents are different from that of commonly used datasets which are composed of either scientific documents or news articles. Quran translation is from religious domain, while SemEval-2010 is from scientific domain.

For the SemEval-2010 and 500N-KPCrowd data sets the ground truth or gold standard is provided for each document within the datasets. Unfortunately, like the datasets for other domains, the gold standard and benchmarking dataset for Quranic domain is not available because to the best of my knowledge, this is the first attempt to use it for the analysis of the key concepts identification algorithms. Also, creating a proper dataset in this domain was out of the scope of the study. Therefore, the easiest way found is to take advantage of the domain experts as they have the knowledge of the field. furthermore, for Quranic dataset we are dependent on domain experts for validation against the ground truth, so, we selected 5 chapters as allowed by the domain experts. To evaluate the results on Quranic dataset a simple procedure is followed. For computing precision and recall, from the domain experts we required to verify the output of the algorithms and identify the true positives, false positives and true negatives. Therefore, they were instructed just to mark the true positives as 1 and false positives as 0 and provide the list all gold standard against each selected of the selected document so that to find the number of true negatives. We did not follow the strict passing criteria for a phrase to be considered as key concept, i.e., the intersection of all lists, rather we set 40% passing criteria meaning that if at least two out of the five experts agree on a key concept then it is considered as key concept. The statistics of selected datasets are given in Table 1.

**Table 1.** Statistics of the datasets used.

Dataset	Domain	Total Number of Docs/Chapters	Avg. Number of Words per Doc/Chapter	Avg. Gold Standard Key Concepts per Doc/Chapter
SemEval-2010	Scientific papers	244	8021.0	15.18
Quranic	Religious Book	114	1469.8	28.25
500N-KPCrowd	News Stories	500	432.73	39.9

**Pre-processing:** Each of the selected algorithm has a pre-processing step to convert the data into a processable form for key concept extraction. In the pre-processing step various tasks are performed. For example, the given document is split into sentences and then into words. Part-of-speech tagging and stemming techniques are applied to obtain part of speech tags and stemmed forms of the words. The filtering of the data is carried out to remove unnecessary words e.g., stop-words and punctuation marks, etc. All these steps are common among the selected algorithms except the Part-of-speech tagging, which is only part of TopicRank.

**Parameters setting:** Table 2 shows the best parameter values for each of the systems. N is the number of extracted key concepts by each system. In KP-Miner [8], Lasf is the least allowable seen frequency, cutoff constant is the total words after that candidates are filtered out, sigma ( $\sigma$ ) and alpha ( $\alpha$ ) are used to compute boosting factor for candidate phrase. In TopicRank [21] similarity threshold is



used to compute similarity between candidate concepts for clustering. Also among the normally used linkage methods for clustering, we select the Average linkage. To mention one point that as the cutoff constant depends on the length of documents in the dataset, so we found it best at a higher value than its original value 400.

*Execution details:* The selected algorithms are first fine-tuned for optimal parameters settings on each dataset, and then with the best settings, results for each of the three selected methods are obtained. The values for sigma ( $\sigma$ ) and alpha ( $\alpha$ ) are set the same as reported by the authors of KPMiner, because they experimentally found the same values best for all datasets they had used. Similarly, in TopicRank we set the same values for the parameters, i.e., similarity threshold and clustering linkage, as reported by the authors. However, because the parameter Lasf, Cutoff constant and N depends on the length of the documents of a given dataset, so we experimentally determined the best values of the parameters for different combination from their range values described in Table 3 on all the selected datasets. The matrix f-measure (described in Section 4.2.1) is used to determine the values, because f-measure is the harmonic mean value of precision and recall that will be high when both the precision and recall are reasonably high.

**Table 2.** Best Parameter values setting on the selected datasets.

Dataset	KP-Miner (Parameters)					TF-IDF (Parameter)	TopicRank (Parameters)		
	N	Lasf	Cutoff Constant	Sigma ( $\sigma$ )	Alpha ( $\alpha$ )	N	N	Similarity Threshold	Clustering Linkage
SemEval-2010	12	3	800	3	2.3	14	20	25%	Average linkage
Quranic	14	4	1000	3	2.3	16	18	25%	Average linkage
500N-KPCrowd	18	3	500	3.0	2.3	16	22	25%	Average linkage

**Table 3.** Various parameters and their range values.

Parameter	Values Range
Sigma ( $\sigma$ )	2.8, 3.0, 3.2, 3.4, 3.6, 3.8
Alpha ( $\alpha$ )	2.2, 2.3, 2.4
Lasf	2–5
Cutoff	400, 600, 800, 1000, 1200, 1400
Clustering Linkage	Single, Complete, Average
N	<25

## 4.2. Performance Analysis

### 4.2.1. Performance Measures

We used the following measures to analyze the performance of the selected methods.

1. **Precision** measures the probability that if a phrase is selected as key concept by an algorithm then it is actually a key concept. It is the proportion of correctly identified key concepts among all retrieved phrases. In keyphrase extraction, usually one would be interested in retrieving top K concepts, so we use Precision at K (P@K).
2. **Recall** measures the probability that if a phrase is key concept then the algorithm will correctly retrieve it. It is the proportion of correctly identified key concepts among all the standard key concepts.
3. **F-measure** There is a tradeoff between precision and recall, if you are interested in extracting all key concepts then recall might be 100% but precision (P@K) will tend to 0%. In converse, if you want to optimize such that each extracted phrase should be really a key concept, then P@K might be 100% but the chances to extract all keyphrases will be close to 0%. Therefore, another measure called F-measure is widely used in information extraction that yields maximum value when

there is balance between precision and recall. A high value of F-measure implies at reasonably high value of both precision and recall [53–55]. F-measure is the harmonic mean of precision and recall:

$$F\text{-Measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

4. **Average Precision (AP)** Precision, Recall and F-measure are single-value metrics that are computed over the whole list of concepts retrieved. However, as keyphrase extraction algorithms retrieve a ranked list of key concepts, so it is desirable to consider the ranking order in which the key concepts are extracted. Therefore, we use in our analysis the measure Average Precision which is a preferred measure for evaluating key concepts extraction algorithms that aims at ranking. Average Precision (AP) is defined as the area under a precision-recall curve. AP is a single-figure quality measure across the recall scores. To be more specific, it is the average of precision computed after each retrieved key concept in the ranked list that is matched in the gold standard. In our case, the following equation is used to calculate AP of the methods [1,56]

$$AP = \frac{1}{R} \sum_{i=1}^{22} r_i \left( \frac{\sum_{j=1}^i r_j}{i} \right) \quad (7)$$

where R represents the total relevant key concepts extracted by the method,  $r_i$  is set to 1 if  $i$ th extracted key concept is relevant, otherwise, set to 0. In the ranked list the key concepts at the top contribute more to the AP than the lower ranked concepts.

5. **Average Multiword Phrases** as mentioned by Nakagawa and Mori [1,57], 85% of keyphrases are normally comprised of multi words. Therefore, we are interested to analyze the performance in terms of multiword phrases extracted by each system. To the best of our knowledge this is the first attempt to compare keyphrase algorithms on this metric. To compute Average Multiword phrases, we count the average number of multi word key concepts that match the gold standard.

After introducing the various performance measures, firstly, we individually analyze the performance of the selected methods on each of the three datasets. We explain the performance in terms of precision-recall curves (see Figure 2). Also, we plot multiword graphs for each system on each dataset, showing the performance in terms of the average number of multiword key concepts extracted by each system during the series of experiments (see Figure 3). The curves are generated by varying, K (1 to 20), the number of key concepts extracted by each system and plotting the best values obtained. In addition, we also explain the effect of numerous factors, included in the ranking formula of each method, on weights assigning to candidate concepts by changing the formula parameters. For instance, in TF-IDF method and KP-Miner we vary the number of documents in the corpus, that in turn changes the IDF (Inverse document frequency) factor. Next, we discuss the overall performance in terms of Average Precision and F-measure.

#### 4.2.2. Individual Performance

In Table 4 the detailed results of the selected algorithms is shown. The significant performance in terms of F-measure at various cut-off points is shown in bold face. The performance at each cut-off point  $N$  is computed using the following equations.

$$\text{Precision}_N = \frac{\sum_{i=0}^n TP_N}{\sum_{i=0}^n (TP_N + FP_N)} \quad (8)$$

$$\text{Recall}_N = \frac{\sum_{i=0}^n TP_N}{\sum_{i=0}^n (TP_N + TN_N)} \quad (9)$$

$$F\text{-measure}_N = \frac{2 \times (\text{Precision}_N + \text{Recall}_N)}{2 \times \text{Precision}_N \times \text{Recall}_N} \quad (10)$$



where  $TP_N$  is total true positives,  $FP_N$  is the total false positives and  $TN_N$  is the number of true negatives at cut-off point  $N$ , and  $n$  is the number of documents.

At the cut-off point  $N = 2$  the total true positives  $TP_N$  are zero on SemEval-2010 dataset which result in 0s values for the corresponding metrics, while on Quranic the total false positives  $FP_N$  are zero at that point for TF-IDF and KP-Miner which result in maximum values for the corresponding metrics. In the coming paras we individually discuss and analyze the performance of each algorithm.

**Table 4.** Performance of the selected algorithms at chosen cut-off values.

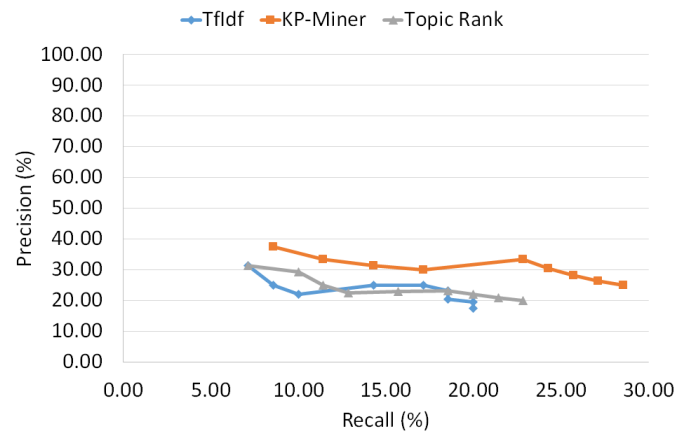
Dataset	TF-IDF				KP-Miner			TopicRank		
	$N$	Precision	Recall	F Score	Precision	Recall	F Score	Precision	Recall	F Score
SemEval-2010	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4	31.25	7.14	11.62	37.50	8.57	<b>13.95</b>	31.25	7.14	11.62
	6	25.00	8.57	12.76	33.33	11.43	<b>17.02</b>	29.17	10.00	14.89
	8	21.88	10.00	13.73	31.25	14.29	<b>19.61</b>	25.00	11.43	15.69
	10	25.00	14.29	18.19	30.00	17.14	<b>21.82</b>	22.50	12.86	16.37
	12	25.00	17.14	20.34	33.33	22.86	<b>27.12</b>	22.92	15.71	18.64
	14	23.21	18.57	20.63	30.36	24.29	<b>26.99</b>	23.21	18.57	20.63
	16	20.31	18.57	19.40	28.13	25.71	<b>26.87</b>	21.88	20.00	20.90
	18	19.44	20.00	19.72	26.39	27.14	<b>26.76</b>	20.83	21.43	21.13
	20	17.50	20.00	18.67	25.00	28.57	<b>26.67</b>	20.00	22.86	21.33
Quranic	2	100.00	3.92	<b>7.55</b>	100.00	3.92	<b>7.55</b>	50.00	1.96	3.77
	4	75.00	5.88	<b>10.91</b>	75.00	5.88	<b>10.91</b>	50.00	3.92	7.27
	6	50.00	5.88	<b>10.53</b>	50.00	5.88	<b>10.53</b>	50.00	5.88	<b>10.53</b>
	8	37.50	5.88	<b>10.17</b>	37.50	5.88	<b>10.17</b>	37.50	5.88	<b>10.17</b>
	10	40.00	7.84	<b>13.11</b>	40.00	7.84	<b>13.11</b>	30.00	5.88	9.84
	12	41.67	9.80	<b>15.87</b>	41.67	9.80	<b>15.87</b>	25.00	5.88	9.52
	14	35.71	9.80	15.38	42.86	11.76	<b>18.46</b>	28.57	7.84	12.31
	16	37.50	11.76	<b>17.91</b>	37.50	11.76	<b>17.91</b>	37.50	11.76	<b>17.91</b>
	18	33.33	11.76	17.39	33.33	11.76	17.39	38.89	13.73	<b>20.29</b>
	20	30.00	11.76	16.90	30.00	11.76	16.90	35.00	13.73	<b>19.72</b>
500N-KPCrowd	2	37.50	3.41	6.25	37.50	3.41	6.25	50.00	4.55	<b>8.33</b>
	4	37.50	6.82	11.54	43.75	7.95	13.46	50.00	9.09	<b>15.38</b>
	6	33.33	9.09	14.29	33.33	9.09	14.29	45.83	12.50	<b>19.64</b>
	8	31.25	11.36	16.67	34.38	12.50	18.33	40.63	14.77	<b>21.67</b>
	10	30.00	13.64	18.75	30.00	13.64	18.75	35.00	15.91	<b>21.88</b>
	12	29.17	15.91	20.59	29.17	15.91	20.59	29.17	15.91	<b>20.59</b>
	14	26.79	17.05	<b>20.83</b>	30.36	13.64	18.82	26.79	17.05	<b>20.83</b>
	16	29.69	21.59	<b>25.00</b>	28.13	20.45	23.68	25.00	18.18	21.05
	18	26.39	21.59	23.75	27.78	22.73	25.00	27.78	22.73	<b>25.00</b>
	20	25.00	22.73	23.81	26.25	23.86	<b>25.00</b>	26.25	23.86	<b>25.00</b>

1. **TF-IDF** The common observation for most of the key concept extraction methods is that by increasing,  $K$ , the number of key concepts predicted by each system, the recall increases while precision decreases. The precision-recall curves (Figure 2) show, that TF-IDF is consistent with this intuition. The overall performance of TF-IDF on SemEval-2010 benchmark dataset is low compared to KP-Miner but matching to TopicRank with slightly high value as shown in Figure 2a. Also, the curve in Figure 3a indicates that the average multiword concepts extracted by the system remains stable at a low of 1.25. In contrast, on Quranic and 500N-KPCrowd datasets the precision-recall curve of TF-IDF shows somewhat overlapping progression with KP-Miner (Figure 2b,c). However, the average multiword key concepts extracted are still not more than 1.25. The reason of low performance could be the fact that *tf-idf* model can potentially result in missing multiword concepts. This make sense because the factor *tf* (term frequency) is dominating than *idf* (Inverse document frequency). The *tf* measures how frequent a word is in a document and nothing can affect this value, whereas *idf* measures how rare a word is across the documents in the corpus and it is dependent on the number of documents,  $N$ , in the corpus and the value of document frequency. Thus, *idf* is affective only if there are more documents in the corpus and document frequency of a word or phrase is low. Based on this argument we can say that, despite the fact that 85% of keyphrases are normally comprised of multi words, single terms will gain

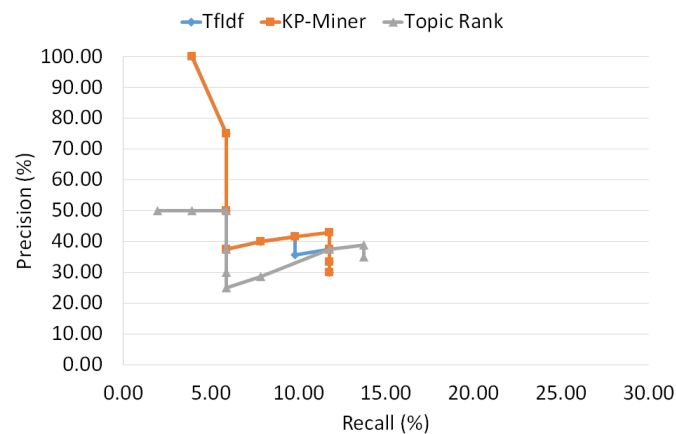
more weights than multiword phrases because it has been found that single word terms occur more frequently as compared to multiword phrases [8]. Therefore, this weakness of *tf-idf* based data-driven approach may result in missing important multiword key concepts, and in turn affect their performance.

2. **KP-Miner** The KP-Miner precision-recall curves (Figure 2) show a similar progression to that of TF-IDF, precision falls when recall raises. The overall performance of KP-Miner on SemEval-2010 dataset is better than both TF-IDF and TopicRank. For all the variations of top K key concepts, the highest scores are achieved by KP-Miner (Figure 2a). We may attribute this to the fact that KP-Miner weighs more to multiword concepts as can be seen in Figure 3a. KP-Miner is based on *tf-idf* model and as discussed earlier that *idf*, which measure the rareness of a phrase, is effective only if there are more documents in the corpus and document frequency of a word or phrase is low. Therefore, because multiword phrases are less frequent and rare across the document corpus, therefore, on SemEval-2010 dataset, where the number of documents in the corpus is higher than Quranic dataset, the multiword concepts may get some effective score. By investigating the other factors that contribute to higher number of multiword keyphrases extracted by KP-Miner, it is found that the author of KP-Miner assumes that compound keyphrases do not occur more frequently compared to single words with in a document set. Based on this assumption the document frequency for multiword key concepts is set to 1, which will result in maximum IDF value, thus giving maximum score to multiword key concepts. We speculate that here KP-Miner is biased towards multiword key concepts. The performance of KP-Miner on Quranic and 500N-KPCrowd dataset supports our argument because in that case the *idf* values of both methods are close to each other, for both single and multiword concepts that result in somewhat overlapping patterns with TF-IDF (Figure 2b,c).
3. **TopicRank** This method exhibits different patterns. While, on SemEval-2010 dataset the performance of TopicRank in terms of precision- recall is close to TF-IDF and lower than KP-Miner, on Quranic dataset its results show unstable behavior (Figure 2b,c). First the precision does not fall as recall rises, then suddenly it falls and recall remains stable at 5.88. After that a gradual increase in precision can be seen. By dipping in depth to determine, why TopicRank performing low and behaves differently in an unstable way on SemEval-2010 and Quranic dataset, we found that the main responsibility lies in the way of generating topics and their weighting. In the first step of identifying candidate concepts, it relies on noun phrases. However, the noun phrases may contain too common and general terms or noise ones [1]. Also, it is not necessary that all concepts must be noun phrases. Verb phrases may also contain important key concepts. For example, in the keyphrase “extracting concepts” “extracting” is verb of type VBG (verb gerund) not NN (noun), but potentially it is similar to “concept extraction”. Similarly, when the key concept “distributed computing” is analyzed the word “distributed” is tagged as verb of type VBN (verb). Therefore, relying only on noun phrases is not enough for key concept extraction. This may result in missing many valuable key concepts. In the next step of making clusters from candidate phrase, it is found that the similarity between candidates is not computed semantically, rather checked lexically with a minimum overlapping threshold value of 25%. This may result in generating topics that group candidates which are lexically similar but semantically opposite. For instance, “supervised machine learning” and “unsupervised machine learning” have lexical similarity but semantically both are opposite concepts. The effect of this will be obvious in the next steps of building graph from the topics and their ranking. Semantically similar key concepts may go to wrong topics, and their co-occurrence weight will be assigned to wrong edges in the graph, thus it may co-relate wrong topics, and ultimately wrong topics may gain higher weights. Therefore, comparing TopicRank with TF-IDF and KP-Miner, we conclude that the co-occurrence based relatedness weighting scheme of TopicRank is uncertain compared to frequency-based weighting scheme of TF-IDF and KP-Miner. Therefore, the same uncertainty can be seen in the unstable results of TopicRank. However, on 500N-KPCrowd dataset it outperforms than its competitors, in terms of precision recall curve (Figure 2c). The reason could be that in 500N-KPCrowd dataset

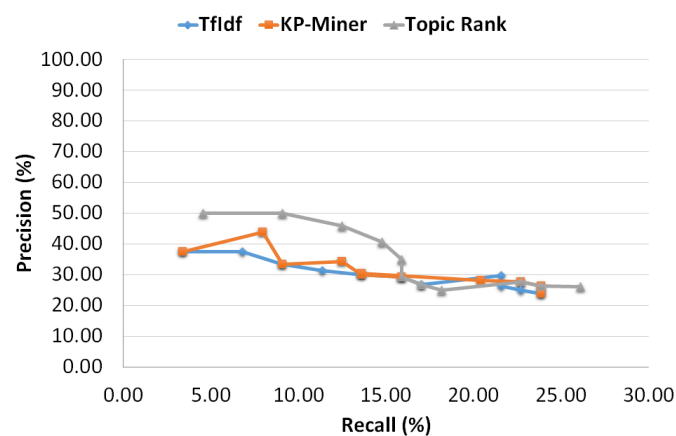
the average number of words per document is very low as compared to the other datasets, in which case the lexical-based similarity may be fruitful that would result in improved precision. Similarly, a gradual increase in the performance can be seen across all the three datasets, in terms of Average Multiword Phrases (Figure 3). this can be attributed to the fact that it does not depend on the frequency-based model tf-idf which is hard to be optimized for multiword phrases.



(a) SemEval-2010.

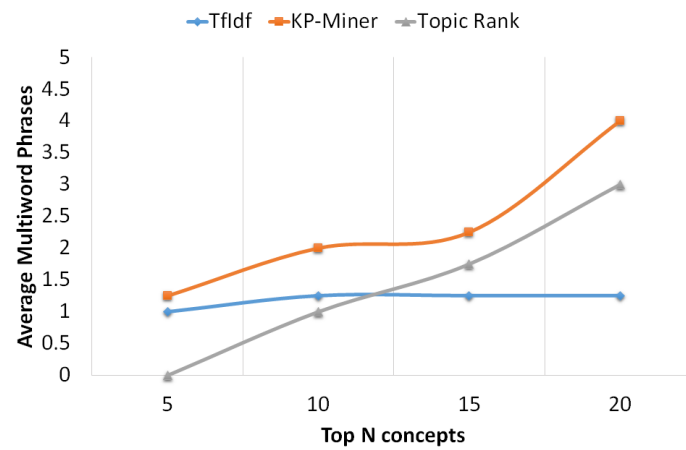


(b) Quranic dataset

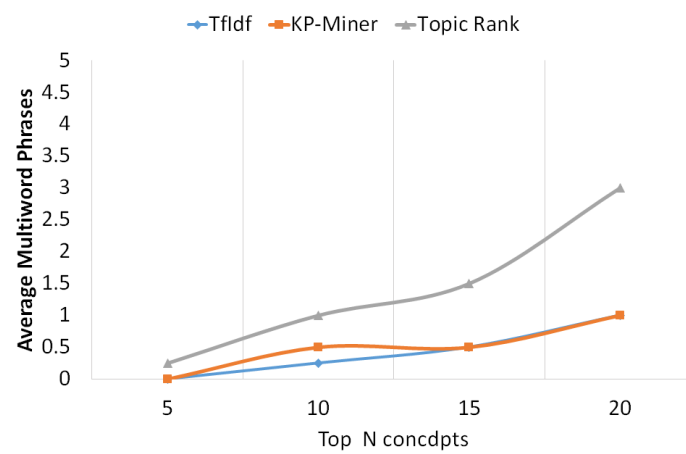


(c) 500N-KPCrowd dataset

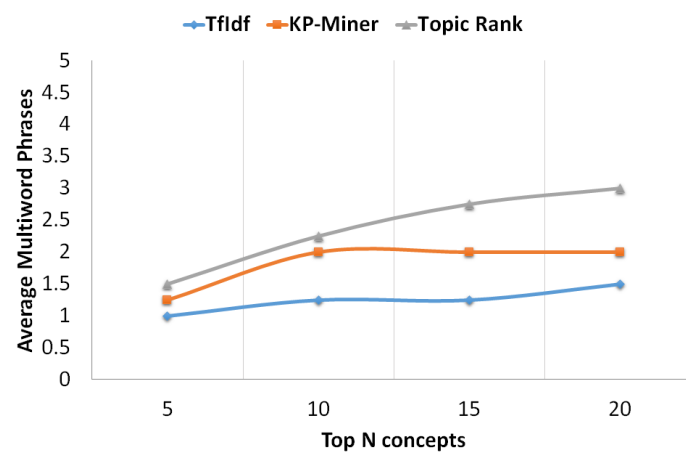
**Figure 2.** Precision and Recall curves for selected datasets.



(a) SemEval-2010.



(b) Quranic dataset



(c) 500N-KPCrowd dataset

**Figure 3.** Average Multiword Phrases curves for selected datasets.

The pattern of precision-recall curves depends on the distribution of key concepts in the dataset. The distribution of key concepts may vary across datasets from different domains. Therefore, a variation in precision-recall curves across different datasets can be seen. However, the common intuition is that precision decreases as recall increases which can be observed in all the precision-recall curves.

### 4.2.3. Overall Performance

We now present the overall performance of the above methods in terms Average Precision (AP), which measures that how early in the ranking list a ranking algorithm fills the position. Table 5 shows that KP-Miner outperforms in terms of AP on SemEval-2010 and Quranic datasets, whereas TopicRank achieve high score on 500N-KPCrowd dataset. Although we have discussed earlier in details the performance of KP-Miner and TopicRank with possible reasons, the fact remains the same that “a good method ranks actual relevant key concepts near the top of the ranking list, while a poor method takes a higher score for precision to reach a higher score for recall” [1].

**Table 5.** Comparison in terms of Average Precision (AP).

Method	AP (%) SemEval-2010	AP (%) Quranic	AP (%) 500N-KPCrowd
KP-Miner	30.59	60	32.06
TopicRank	24.08	42.49	35.34
TF-IDF	24.4	58.83	30.41

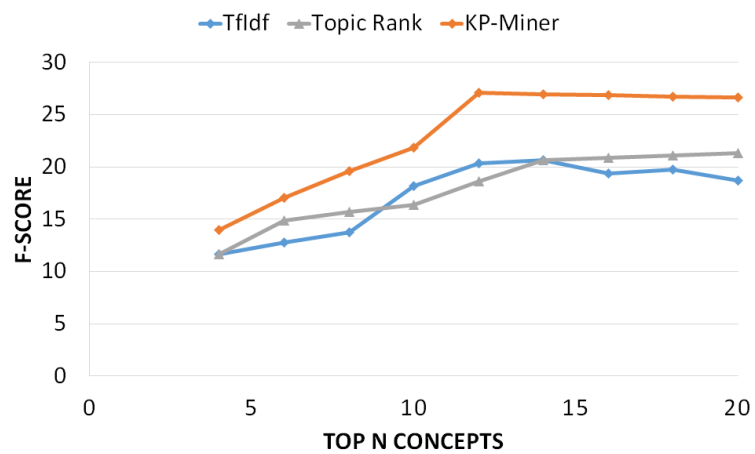
We now evaluate the overall performance of the selected methods in terms of F-Measure on the selected datasets. Figure 4 indicates the F-measure curves for the methods on each of the three datasets, connecting the F-measure scores at various positions. A common observation on both datasets is that as the number of top N concepts increases the F-measure score also increases, reaching a maximum value. On SemEval-2010 dataset the KP-Miner score becomes larger with the increase in ranking position, as compared to TF-IDF and TopicRank (Figure 4a). However, on Quranic and 500N-KPCrowd datasets KP-Miner shows overlapping pattern to that of TF-IDF (Figure 4b,c), and the reason is the same as discussed earlier that KP-Miner is also based on *tf-idf* model. Therefore, when the *idf* score for both methods is close to each other, then slight difference remains between them.

In the precision-recall curves, the precision at N recall values ( $P@N$ ) is computed. Technically, the cut-off value for N should be selected such a point that after that point the f-measure value drops significantly for all the comparing algorithms on the dataset. Because one dataset is different from other, therefore, the value for cut-off N may be different from one dataset to another (see Figure 4). We observed this difference during our experiments, based on which we set the cut-off values.

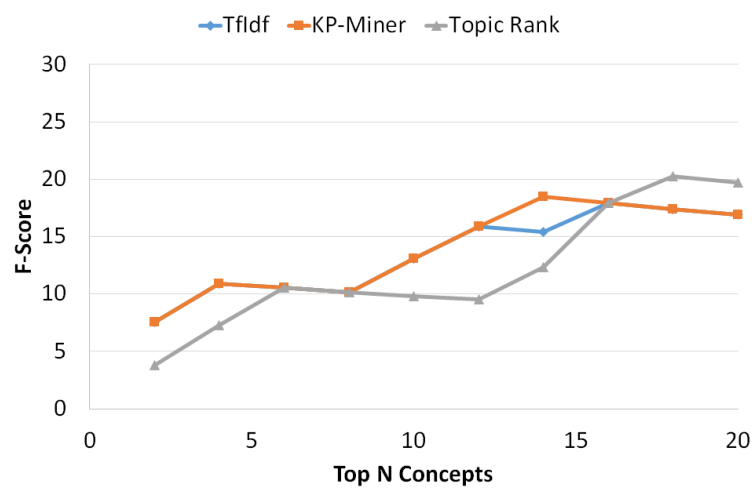
The maximum F-measure score obtained for each method is shown in Table 6. In terms of F-measure KP-Miner outperforms on SemEval-2010 when the number of extracted concepts is 12, while on Quranic and 500N-KPCrowd datasets TopicRank achieves maximum score when number of extracted concepts is 18 and 20 respectively. A good method reaches a higher F-measure score near the top in the ranking list, in converse a poor method reaches a higher score near the end of the list.

**Table 6.** Comparison in terms of F-Measure.

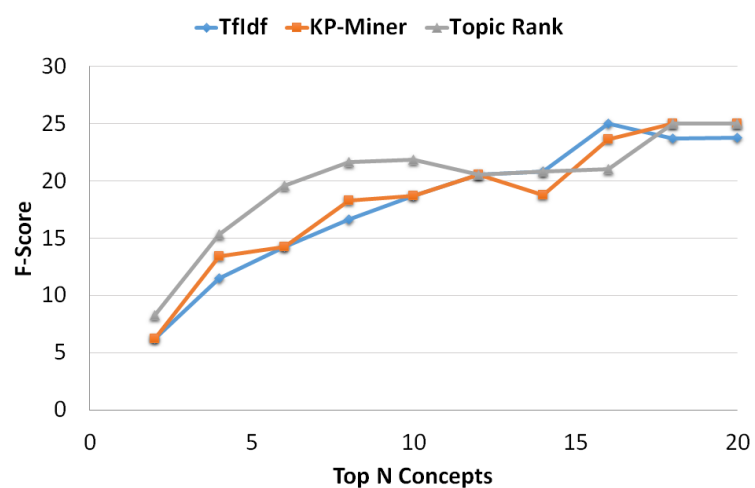
Method	F-Measure (%) SemEval-2010	F-Measure (%) Quranic	F-Measure (%) 500N-KPCrowd
KP-Miner	27.12	18.46	25
TopicRank	21.33	20.29	26.14
TF-IDF	20.63	17.91	25



(a) SemEval-2010.



(b) Quranic dataset



(c) 500N-KPCrowd dataset

**Figure 4.** F-measure curves for selected datasets.



### 4.3. Error Source Analysis

Based on the above performance analysis, we now present error analysis with objective to quantitatively describe the major error sources that contribute to precision or recall errors of the algorithms, that will also provide future work directions. For this purpose, we manually analyzed the systems output on 30 randomly selected documents from SemEval-2010 dataset and 5 chapters from Quranic dataset and 40 documents from 500N-KPCrowd dataset. The heterogeneity in the number of selected documents is for two reasons. First, the number of documents are selected based on the size of the documents in the dataset, i.e., for larger size documents, we selected fewer documents. Second, for SemEval-2010 and 500N-KPCrowd the ground truth or gold standard is provided for each document within the dataset and for Quranic dataset, as we were dependent on domain experts for validation against the ground truth, therefore, we selected 5 chapters as allowed by the domain experts. We determined the proportion of the total number of errors of particular type to the accumulated number of false positives for each algorithm on the selected data. Hassan and NG [13] described four kind of errors commonly made by keyphrase extraction systems namely, overgeneration errors, infrequency errors, redundancy errors, and evaluation errors. However, adding to them, we identified three more major categories i.e., syntactical errors, frequency errors and semantical errors that are made by the selected methods. In Table 7 we summarize the results of the error source analysis. Inline to the objective of this preliminary study, the table does not aim at comparing the algorithms rather than to provide the proportions of the different kind of errors found in the total false positives. Therefore, the best statistical method that suits to our case is to give confidence level to the proportions instead of performing significance test. The results are presented with 95% confidence interval. In future study, this result will help us to develop a robust solution for key concepts identification and to overcome the different kind of errors. In the subsequent paras, we describe each of the three categories of errors.

**Syntactical Errors:** These are precision errors that occurs when a system extract keyphrases that are syntactically incorrect. In statistical n-gram-based methods this kind of errors ranges from 18 to 25% as shown in Table 7, because they can select grammatically, wrong combination of words. For example, the keyphrase “querying multiple”, extracted from SamEval-2010 is syntactically incorrect. however, the correct one is “querying multiple registries”.

**Frequency Errors:** These are major precision errors that occurs when the extracting system results in more single word terms than multiword keyphrases due to the fact that single term concepts more frequently occur than multiword concepts. we analyzed in the previous section that in statistical frequency-based ranking algorithms the single word keyphrases achieve higher scores than multi word phrases, although, in some algorithms multi word concepts are biasedly given higher weights. Our error analysis supports this argument, as we found through error analysis by manually analyzing the output files that on average about 85% of the extracted terms are single words, out of which about 60% are non-key concepts. This high false positive rate of single words contributes to 40 to 45% of overall errors.

**Semantic Errors:** This major kind of recall errors are found in results of TopicRank, contributing to almost 28% of overall errors. This error occurs when the system fails to retrieve concepts that are lexically similar to extracted concepts but semantically opposite. For example, the key concepts “UDDI registries” and “proxy registries” are lexically similar but semantically different. However, TopicRank cluster them under same topic based on lexical similarity.

The errors identified in this study could be addressed at different levels of key concepts identification. For example, the syntactical errors can be best handled in candidate selection step, using parsing techniques and extracting meaningful structures. The semantic errors can be overcome at topic identification level, using semantic-based clustering techniques or topic models e.g., Latent Dirichlet Allocation (LDA) or N-gram topical model (TNG) [58]. Similarly, the frequency errors are related to syntactical errors that can be reduced if the algorithm can produce a comprehensive and meaningful list of candidate phrases.

Another, important aspect to discuss is that how the error sources identified in this study are related to the previously identified error sources [13]. The semantic errors occur when lexically related candidate phrases are clustered under same topic and both of them are key concepts, but the system retrieves one, while the redundancy errors occur when two semantically related candidate phrases are grouped under the same topic and one of them is key concept, but the system retrieves both. The frequency and infrequency errors are closely related, having slight difference. The infrequency errors are a general category of the recall errors that occur when the system fails to retrieve a key concept due to the fact that it is infrequent, in converse the frequency errors are precision errors that occur when the system retrieves more single words due to the fact that they are frequent, but they are not key concepts. The third kind of error source identified in this study i.e., syntactical errors are precision errors are that occurs when the extracted candidate phrases are syntactically incorrect.

**Table 7.** Summary of the Error Source Analysis.

Algorithm	Total False Positives	Error Source	95% Confidence Interval (%)	Type
TF-IDF	1175	Frequency errors	$45 \pm 2.85$	Precision errors
		Syntactical errors	$25 \pm 2.48$	
KP-Miner	1110	Frequency errors	$40 \pm 2.88$	
		Syntactical errors	$18 \pm 2.26$	
Topic Rank	1135	Semantical errors	$28 \pm 2.62$	Recall errors

## 5. Conclusions

In this study initially, we have conducted a brief survey of keyphrase extraction algorithms and categorized them describing the necessary details and limitation of different approaches. After that, we conducted an empirical analysis of three state-of-the-art unsupervised data driven key concept extraction methods on three datasets from different domains. We draw several conclusions from our analysis. (1) By using statistical frequency-based approach for key concepts ranking, the single word concepts achieve higher scores than multi word concepts that result in the major precision errors called Frequency errors ranging from  $40 \pm 2.88$  to  $45 \pm 2.85\%$  of overall errors as shown in Table 7. There could be three factors that contribute in higher scores of single terms. First, single term concepts more frequently occur than multiword concepts. Second, the term frequency factor *tf* in frequency-based measure (*tf-idf*) is dominant than *idf*. Third, multiword key concepts are highly dependent on *idf* factor which is sensitively affected by total number of documents in corpus. (2) the statistical n-gram-based approaches for candidate selection may select grammatically, wrong combination of words that may result in precision errors called Syntactical errors, this kind of errors ranges from  $18 \pm 2.26$  to  $25 \pm 2.48\%$ . (3) Using lexical similarity for clustering candidates under different topics may result in recall errors called Semantic errors that contributes to  $28 \pm 2.62\%$  of overall errors. Finally, as we discussed earlier, that the way of key concept candidate's selection and their ranking may have a strong impact on overall key concepts extraction process, therefore, in future investigating alternative methods that give appropriate weight to multiword key concepts and consider semantic similarity for grouping words under different topics may be worthwhile.

To overcome the shortcomings of existing systems, in future an integrated solution is needed. Parsing techniques can be used in the pre-processing step of the solution to produce a comprehensive list of candidate phrases from the input text documents, that may reduce syntactical and frequency errors. Various topic models or clustering techniques can be used to find topics based on semantic relatedness, that may address semantical errors.

**Author Contributions:** M.A. conducted the survey, performed experimental analysis and paper writeup. A.b.M.S. and S.J.A.K. conceived the idea and supervised this work. I.U. assisted in designing experimental setup and paper writeup. All authors contributed to this paper.

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

## Appendix A. Summary of Different Keyphrase Extraction Methods

**Table A1.** Summary of different keyphrase extraction methods.

Source	Category	Approach Used	Techniques Used	Remarks	Limitations
KEA [14]	Supervised	Statistical and structural-based	Term Frequency, Phrase Position	Language Independent. Relying only on statistical information may result in missing important multiword phrases.	Require manually annotated quality training set. Training process make them domain dependent.
GenEx [10]	Supervised	Statistical and structural-based	Term Frequency, Phrase Position		
[15]	Supervised	Statistical and linguistic-based	Lexical features e.g., collection frequency, part-of-speech tags, Bagging technique		
KEA++ [33]	Supervised	Statistical and linguistic based	NLP techniques, Using Thesaurus		
[34]	Supervised	Statistical and linguistic-based	Distribution information of candidate phrase	Extension of KEA. Language dependent, may require domain knowledge and expertise in language. Glossaries or auxiliary structures are useful however, they require extensive human efforts in definition of terms and terminology standardization.	
[16]	Supervised	Statistical and linguistic based	Integration of Wikipedia		
[32]	Supervised	Statistical and linguistic-based	Structural features e.g., presence of a phrase in specific section. Lexical features e.g., presence of phrase in Wordnet or Wikipedia. Bagged decision tree		
[31]	Supervised	Statistical and linguistic based	Statistical and linguistic features e.g., tf-idf, BM25, POS		
[30]	Supervised	Statistical and linguistic based	Features based on citation network information along with traditional features		
[35]	Un-Supervised	Statistical-based	tf-idf (term frequency-inverse document frequency). Topic proportions	Target process is the ranking of candidate phrases. Language Independent	Relying only on statistical information may result in missing important multiword keyconcepts due to higher weights to single terms. Semantics free extraction
[38]	Un-Supervised	Statistical-based	tf-idf (term frequency-inverse document frequency). Topic proportions		
[27]	Un-Supervised	Statistical-based	tf-idf (term frequency-inverse document frequency). Topic proportions		
[37]	Un-Supervised	Statistical-based	tf-idf (term frequency-inverse document frequency). Topic proportions		
[8]	Un-Supervised	Statistical-based	Tf-idf, boosting factor		

Table A1. Cont.

Source	Category	Approach Used	Techniques Used	Remarks	Limitations
[40]	Un-Supervised	Linguistic or syntactical information-based	Considers Part-of speech tags other than noun and adjectives		Language dependent, may require domain knowledge and expertise in language. Glossaries or auxiliary structures require extensive human efforts in definition of terms and terminology standardization.
[39]	Un-Supervised	Linguistic or syntactical information-based	Creates a database containing semantically related keyphrases		
CFinder [1]	Un-Supervised	Statistical, syntactical and structural information-based	Statistical and structural information. Domain-specific knowledge		
Topic-biased PageRank [47]	Un-Supervised	Topical clustering-based	Topic models		Extension of topic-biased PageRank
[46]	Un-Supervised	Topical clustering-based	Topic models. Decomposing documents into multiple topics		
TopicRank [21]	Un-Supervised	Topical clustering-based	Clustering techniques to group candidate phrases into topics		
TextRank [22]	Un-Supervised	Graph-based ranking	PageRank algorithm	Adjacent words are used to build the graph	Prefer single words as nodes of the graph, thus may result in missing important multiword keyphrases. Does not guarantee covering all topics.
SingleRank [23]	Un-Supervised	Graph-based ranking	Co-occurrence window of variable size $w \geq 2$ . lexically-similar neighboring documents	Extension of TextRank	
ExpandRank [24]	Un-Supervised	Graph-based ranking	Co-occurrence window of variable size $w \geq 2$ . lexically-similar neighboring documents		
[42]	Un-Supervised	Graph-based ranking	Citation network information	Extension of ExpandRank	
[43]	Un-Supervised	Graph-based ranking	Centrality measures e.g., node degree, closeness, and clustering coefficient		
[44]	Un-Supervised	Graph-based ranking	WordNet information	WordNet is used to find semantic relationship between words	
SGRank [41]	Un-Supervised	Graph-based ranking	Statistical Heuristics e.g., Tf-Idf, First position of a keyphrase in a document		Finds semantic relatedness between words in a graph
[45]	Un-Supervised	Graph-based ranking	Word embedding vectors		

## References

1. Kang, Y.B.; Haghighi, P.D.; Burstein, F. CFinder: An intelligent key concept finder from text for ontology development. *Expert Syst. Appl.* **2014**, *41*, 4494–4504. [[CrossRef](#)]
2. Noy, N.F.; McGuinness, D.L. *Ontology Development 101: A Guide to Creating Your First Ontology*; Stanford University: Stanford, CA, USA, 2001.
3. Cimiano, P.; Völker, J. A framework for ontology learning and data-driven change discovery. In Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB), Lecture Notes in Computer Science, Alicante, Spain, 15–17 June 2005; Springer: Berlin, Germany, 2005; Volume 3513, pp. 227–238.
4. Jiang, X.; Tan, A.H. CRCTOL: A semantic-based domain ontology learning system. *J. Assoc. Inf. Sci. Technol.* **2010**, *61*, 150–168. [[CrossRef](#)]
5. Li, Q.; Wu, Y.F.B. Identifying important concepts from medical documents. *J. Biomed. Inform.* **2006**, *39*, 668–679. [[CrossRef](#)] [[PubMed](#)]
6. Tonelli, S.; Rospocher, M.; Pianta, E.; Serafini, L. Boosting collaborative ontology building with key-concept extraction. In Proceedings of the 2011 Fifth IEEE International Conference on Semantic Computing (ICSC), Palo Alto, CA, USA, 18–21 September 2011; pp. 316–319.
7. Aman, M.; Bin Md Said, A.; Kadir, S.J.A.; Baharudin, B. A Review of Studies on Ontology Development for Islamic Knowledge Domain. *J. Theor. Appl. Inf. Technol.* **2017**, *95*, 3303–3311.
8. El-Beltagy, S.R.; Rafea, A. KP-Miner: A keyphrase extraction system for English and Arabic documents. *Inf. Syst.* **2009**, *34*, 132–144. [[CrossRef](#)]
9. Englmeier, K.; Murtagh, F.; Mothe, J. Domain ontology: automatically extracting and structuring community language from texts. In Proceedings of the International Conference Applied Computing (IADIS), Salamanca, Spain, 18–20 February 2007; pp. 59–66.
10. Turney, P.D. Learning algorithms for keyphrase extraction. *Inf. Retr.* **2000**, *2*, 303–336. [[CrossRef](#)]
11. Hulth, A.; Megyesi, B.B. A study on automatically extracted keywords in text categorization. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, 17–18 July 2006; Association for Computational Linguistics: Stroudsburg, PA, USA, 2006; pp. 537–544.
12. Hasan, K.S.; Ng, V. Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Beijing, China, 23–27 August 2010; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010; pp. 365–373.
13. Saidul, H.K.; Vincent, N. 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, 22–27 June 2014; Volume 1, pp. 1262–1273.
14. Frank, E.; Paynter, G.W.; Witten, I.H.; Gutwin, C.; Nevill-Manning, C.G. Domain-specific keyphrase extraction. In Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 99), Stockholm, Sweden, 31 July–6 August 1999; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1999; Volume 2, pp. 668–673.
15. 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; Association for Computational Linguistics: Stroudsburg, PA, USA, 2003; pp. 216–223.
16. 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; Association for Computational Linguistics: Stroudsburg, PA, USA, 2009; Volume 3, pp. 1318–1327.
17. Turney, P.D. Coherent keyphrase extraction via web mining. *arXiv* **2003**, arXiv:cs/0308033.
18. Nam, K.S.; Olena, M.; Min-Yen, K.; Timothy, B. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In Proceedings of the 5th International Workshop on Semantic Evaluation, Los Angeles, CA, USA, 15–16 July 2010; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010; pp. 21–26.
19. Kim, S.N.; Medelyan, O.; Kan, M.Y.; Baldwin, T. Automatic keyphrase extraction from scientific articles. *Lang. Resour. Eval.* **2013**, *47*, 723–742. [[CrossRef](#)]

20. 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; Association for Computational Linguistics: Stroudsburg, PA, USA, 2003; Volume 18, pp. 33–40.
21. Bougouin, A.; Boudin, F.; Daille, B. Topicrank: Graph-based topic ranking for keyphrase extraction. In Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP), Nagoya, Japan, 14–19 October 2013; pp. 543–551.
22. Mihalcea, R.; Tarau, P. TextRank: Bringing Order into Text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; Volume 4, pp. 404–411.
23. Wan, X.; Xiao, J. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, Chicago, IL, USA, 13–17 July 2008; Volume 8, pp. 855–860.
24. Xiaojun, W.; Jianguo, X. CollabRank: towards a collaborative approach to single-document keyphrase extraction. In Proceedings of the 22nd International Conference on Computational Linguistics, Manchester, UK, 18–22 August 2008; Association for Computational Linguistics: Stroudsburg, PA, USA, 2008; Volume 1, pp. 969–976.
25. Wan, X.; Yang, J.; Xiao, J. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, 25–27 June 2007; Volume 7, pp. 552–559.
26. Zha, H. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, 11–15 August 2002; ACM: New York, NY, USA, 2002; pp. 113–120.
27. 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 Volume 1–Volume 1, Singapore, 6–7 August 2009; Association for Computational Linguistics, 2009; pp. 257–266.
28. Matsuo, Y.; Ishizuka, M. Keyword extraction from a single document using word co-occurrence statistical information. *Int. J. Artif. Intell. Tools* **2004**, *13*, 157–169. [[CrossRef](#)]
29. Boudin, F. Pke: An open source python-based keyphrase extraction toolkit. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, Osaka, Japan, 11–16 December 2016; pp. 69–73.
30. Caragea, C.; Bulgarov, F.A.; Godea, A.; Gollapalli, S.D. Citation-Enhanced Keyphrase Extraction from Research Papers: A Supervised Approach. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Volume 14, pp. 1435–1446.
31. Chuang, J.; Manning, C.D.; Heer, J. “Without the Clutter of Unimportant Words”: Descriptive keyphrases for text visualization. *ACM Trans. Comput. Hum. Interact.* **2012**, *19*, 19. [[CrossRef](#)]
32. Lopez, P.; Romary, L. HUMB: Automatic key term extraction from scientific articles in GROBID. In Proceedings of the 5th International Workshop on Semantic Evaluation, Los Angeles, CA, USA, 15–16 July 2010; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010; pp. 248–251.
33. Medelyan, O.; Witten, I.H. Thesaurus based automatic keyphrase indexing. In Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, Chapel Hill, NC, USA, 11–15 June 2006; ACM: New York, NY, USA, 2006; pp. 296–297.
34. Nguyen, T.; Kan, M.Y. Keyphrase extraction in scientific publications. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*; Springer: Berlin, Germany, 2007; pp. 317–326.
35. Barker, K.; Cornacchia, N. Using noun phrase heads to extract document keyphrases. In Proceedings of the Canadian Society for Computational Studies of Intelligence, Montréal, QC, Canada, 14–17 May 2000; pp. 40–52.
36. 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 (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 1105–1115.
37. Liu, F.; Pennell, D.; Liu, F.; Liu, Y. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In Proceedings of the Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Boulder, CO, USA, 1–3 June 2009; Association for Computational Linguistics: Stroudsburg, PA, USA, 2009; pp. 620–628.



38. Zhang, Y.; Milios, E.; Zincir-Heywood, N. A comparative study on key phrase extraction methods in automatic web site summarization. *J. Dig. Inf. Manag.* **2007**, *5*, 323.
39. Adar, E.; Datta, S. Building a scientific concept hierarchy database (schbase). In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 27–31 July 2015; pp. 606–615.
40. Le, T.T.N.; Le Nguyen, M.; Shimazu, A. Unsupervised Keyphrase Extraction: Introducing New Kinds of Words to Keyphrases. In Proceedings of the Australasian Joint Conference on Artificial Intelligence, Hobart, TAS, Australia, 5–8 December 2016; Springer: Berlin, Germany, 2016; pp. 665–671.
41. Danesh, S.; Sumner, T.; Martin, J.H. SGRank: Combining Statistical and Graphical Methods to Improve the State of the Art in Unsupervised Keyphrase Extraction. In Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics, Denver, CO, USA, 4–5 June 2015; pp. 117–126.
42. Gollapalli, S.D.; Caragea, 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.
43. Lahiri, S.; Choudhury, S.R.; Caragea, C. Keyword and keyphrase extraction using centrality measures on collocation networks. *arXiv* **2014**, arXiv:1401.6571.
44. Martinez-Romo, J.; Araujo, L.; Duque Fernandez, A. SemGraph: Extracting keyphrases following a novel semantic graph-based approach. *J. Assoc. Inf. Sci. Technol.* **2016**, *67*, 71–82. [CrossRef]
45. Wang, R.; Liu, W.; McDonald, C. Corpus-independent generic keyphrase extraction using word embedding vectors. In Proceedings of the Software Engineering Research Conference, Las Vegas, NV, USA, 21–24 July 2014; Volume 39.
46. Liu, Z.; Huang, W.; Zheng, Y.; Sun, M. Automatic keyphrase extraction via topic decomposition. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Honolulu, HI, USA, 25–27 October 2008; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010; pp. 366–376.
47. Haveliwala, T.H. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Trans. Knowl. Data Eng.* **2003**, *15*, 784–796. [CrossRef]
48. Luhn, H.P. A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Dev.* **1957**, *1*, 309–317. [CrossRef]
49. Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *J. Doc.* **1972**, *28*, 11–21. [CrossRef]
50. Quran English Translation. 2016. Available online: <http://tanzil.net/trans/> (accessed on 21 March 2017).
51. Ouda, K. QuranAnalysis: A Semantic Search and Intelligence System for the Quran. Master's Thesis, University of Leeds, Leeds, UK, 2015.
52. 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. *arXiv* **2013**, arXiv:1306.4886.
53. Rijsbergen, C.J.V. *Information Retrieval*, 2nd ed.; Butterworth-Heinemann: Newton, MA, USA, 1979.
54. Lewis, D.D. Evaluating and optimizing autonomous text classification systems. In Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA, 9–13 July 1995; ACM: New York, NY, USA, 1995; pp. 246–254.
55. Turney, P.D. Extraction of keyphrases from text: Evaluation of four algorithms. *arXiv* **2002**, arXiv:cs/0212014.
56. Turpin, A.; Scholer, F. User performance versus precision measures for simple search tasks. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA, 6–10 August 2006; ACM: New York, NY, USA, 2006; pp. 11–18.
57. Nakagawa, H.; Mori, T. A simple but powerful automatic term extraction method. In *Proceedings of the COLING-02 on COMPUTERM 2002: Second International Workshop on Computational Terminology- Volume 14*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 1–7.
58. Wang, X.; McCallum, A.; Wei, X. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In Proceedings of the Seventh IEEE International Conference on Data Mining, Omaha, NE, USA, 28–31 October 2007; pp. 697–702.

