Rapid and Robust Ranking of Text Documents in a Dynamically Changing Corpus

Byung-Hoon Park[*], Nagiza F. Samatova[*,+,¥], Rajesh Munavalli[*], Ramya Krishnamurthy[*], Houssain Kettani[**], and Al Geist[*]
[*]*CSM Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831*
[+]*Computer Science Department, North Carolina State University, Raleigh, NC 27695*
[**]*ECECS Department, Polytechnic University of Puerto Rico, San Juan, PR 00919*
[¥]*Corresponding author: samatovan@ornl.gov*

Abstract

Ranking documents in a selected corpus plays an important role in information retrieval systems. Despite notable advances in this direction, with continuously accumulating text documents, maintaining up-to-date ordering among documents in the domains of interest is a challenging task. Conventional approaches can produce an ordering that is only valid within a given corpus. Thus, with such approaches, ordering should be completely redone as documents are added to or deleted from the corpus. In this paper, we introduce a corpus-independent framework for rapid ordering of documents in a dynamically changing corpus. Like in many practical approaches, our framework suggests utilizing a similarity measure in some metric space indicating the degree of relevance of a document to the domain of interest. However, unlike in corpus-dependent approaches, the relevance score of a document remains valid with changes being introduced into the corpus (insertion of new documents, for example), thus allowing a rapid ordering within the corpus. This paper particularly details a statistical approach to compute such relevance scores.

1. Introduction

With the increasingly available avalanche of digital documents, fast identification of relevant documents of interest has become an indispensable task. For a statically defined corpus, ranking documents based on their relevancy to the topic of interest has received a lot of attention in literature. When the corpus is dynamically changing, however, most methods require complete re-ordering of documents, as their underlying scoring functions rely on global, or corpus-dependent, statistics. Instant ordering of text documents in a dynamic corpus is of tremendous importance in many practical applications such as identification of potential adversary attacks. Yet, a methodology for robust and accurate statistical scoring of documents in a dynamic corpus is lacking, and is the focus of this paper.

In many practical cases, a topic of interest can be represented as a set of keyphrases (or keywords) [7, 8]. The initial set of keyphrases can be manually suggested, or systematically found from a collection of hand-selected documents. For more accurate representation of the topic, instead of being treated equally, the importance of each keyphrase should be assessed based on its degree of relevancy to the topic. Then, a topic can be represented as a vector of scores. In other words, a topic is a point in a metric space, where each keyphrase is a dimension and the score associated with it is the coordinate. Likewise, a document in a corpus is mapped to a vector in the same metric space by assigning relevancy score to each keyphrase. Ordering of documents in a corpus is then achieved by comparing distances or angles between document vectors and the topic in the metric space.

Given a corpus of related documents, the importance of a keyphrase in the document can be assessed based on simple statistics like Term Frequency (TF) [1] and Inverse Document Frequency (IDF) [2-4], or based on more elaborate similarity functions [5, 6]. Then each document is represented by scores of keyphrases that are computed within the context of the document. However, the ordering is only valid within the initially selected corpus; the ranking of a new document outside the original corpus cannot be determined unless the ranking procedure is completely redone. Considering the number of documents that become newly available or obsolete everyday, maintaining up-to-date ordering among the documents within a domain of interest has cast both practical and technical challenges in information retrieval systems.

In this paper, we propose a corpus-free approach for rapid ordering of documents and discuss its appropriateness. Unlike scoring schemes for corpus-dependent ordering, where scores are computed within the context of the corpus, our scheme assesses the score of a keyword for each document independently. Since assessment of keywords is done exclusively for each document, scores of keywords for the given document are invariant. For a set of scores from a document to be compared with one from another document, a score for each keyphrase is transformed into a universal scale. Hence, no additional assessment of keyphrases is necessary even when a new set of documents is added for consideration.

Robust and accurate estimation and transformation of a keyphrase score is the key to the successful deployment of the proposed ordering scheme. This paper particularly details a scoring scheme that utilizes the chi-square value in co-occurrence distribution with frequent terms in a document, and z-scale transformation of it. The rest of the paper is organized as follows. In Section 2, we describe background of our study by reviewing related works in the fields of keyphrase extraction and document ranking. In Section 3, we introduce our framework for rapid ordering of documents. In Section 4, we report our empirical evaluation of the proposed work, especially the scoring and transformation scheme. Finally, with a discussion on future direction, Section 5 concludes the paper.

2. Background

Our document ranking system largely depends on robust assessment of relevancy scores for keyphrases. Keyword extraction is an important prerequisite step in many information retrieval tasks like text clustering, classification, automatic text summarization, etc. In this section, we review keyphrase extraction algorithms. In particular, we describe how different keyphrase extraction approaches assign relevancy score to a candidate keyphrase, and discuss their appropriateness for ordering of documents. A high-level description of document ranking algorithms is also presented. Throughout the paper, words “keyphrase” and “keyword” as well as “term” and “phrase” will be used interchangeably.

2.1. Keyphrase Extraction

Algorithms for keyphrase extraction can be classified into two broad categories: corpus-dependent and corpus-independent approaches. While the former requires a large collection of documents and predetermined keyphrases to build a prediction model, the latter directly sifts keyphrases from a document without any previous or background information. In this sense, they are often contrasted as supervised and unsupervised learning approaches. Generally it is accepted that corpus-dependent approaches yield better performance. However, a corpus-dependent prediction model is practically restricted to a single domain, thus the quality of extracted keyphrases from a new document of unknown domain is not always guaranteed. In this regard, corpus-independent (or domain-independent) approaches may find many practical applications.

Corpus-dependent keyphrase extraction algorithms are mainly based on a set of features extracted from a training corpus. Specifically, a potential key phrase is mapped to the selected feature space, where various machine learning or statistical techniques are applied to distinguish keyphrases from non-keyphrases. The most widely accepted features are Term Frequency (TF), Inverse Document Frequency (IDF), and location in the document. TF and IDF denote relative importance (or weight) of terms in a document. A term is assigned a high TF and IDF score if it occurs frequently in a given document (TF), yet rarely in all other documents (IDF). These features have been successfully adopted by Naïve Bayes-based classifiers like KEA [9], where TF and IDF are combined into a single feature TF-IDF, $tf_{i,j} \times \log(\frac{N}{df_i})$. Here $tf_{i,j}$, df_i, and N denote frequency of term i in document j, the number of documents containing term i, and the total number of documents in the corpus, respectively. Although corpus-dependent keyphrase extraction algorithms reportedly produce improved performance, they are domain-specific, and cannot be generalized unless training is redone with a new corpus set. Consequently, ranking documents using keyphrases thus obtained is only valid with respect to the given corpus.

Unlike corpus-dependent algorithms, corpus-independent algorithms do not require a domain-specific training corpus. Keyphrases are identified solely based on local context of the input document. Most approaches in this direction are based on the assumption that keyphrases have a distinctive co-occurrence pattern with other terms in the same document. For example, PMI-IR [7, 10] searches for phrases that tend to co-occur in the same document based on point-wise mutual information [11] measure for co-occurrence [12]. PMI-IR assigns the weight to a phrase by its accompanying phrases. On the other hand, a number of algorithms [13, 14] consider the statistical unusualness of a phrase in its co-occurrence pattern with frequent terms. With this strategy, a phrase is given a high weight if it frequently co-occurs with a small number of frequent terms, but less frequently co-occurs with the rest of the frequent terms. The most widely used measures for this are chi-square and Kullback-Liebler distance.

2.2. Document Ordering

Approaches to ranking documents can be classified into two categories: similarity measure-based and graph connectivity-based. While the former utilizes similarity functions that are defined in terms of word frequency-related features, the latter assigns order to a document according to its inter-relationship with other documents. Widely used similarity functions include cosine measure [5] and Okapi BM25 [6]. Interestingly, a genetic programming approach has been applied to find an optimal similarity function with respect to the document collection [15]. Graph-based ranking algorithms represent a document as a vertex and inter-document relationship as an edge. The importance of each vertex is determined by recursively spreading out and adjusting weights of the entire graph in a global fashion. Intuitively, a vertex is associated with a high weight if it is adjacent with many high-weighted vertices. Examples of graph-based ranking algorithms include HITS [16], PageRank [17], TextRank [18], and LexRank [19].

Like corpus-dependent keyphrase extraction algorithms, currently available document ranking algorithms only assign *relative* order to documents within the corpus. Our document ranking framework utilizes corpus-independent keyphrase extraction algorithms. However, weights of two keyphrases extracted from different documents are not generally comparable. In the following section, we introduce a method that addresses this limitation.

3. Rapid Ordering of Documents

Our document ranking framework utilizes the co-occurrence pattern of each keyphrase in a document with the most frequently occurring phrases in the same document. More specifically, the weight of each keyphrase is assigned by computing its chi-square value with respect to the occurrence distribution of frequent phrases as found in [20, 21]. Then weights of keyphrases in a document are transformed so that they are comparable with weights of keyphrases extracted from different documents.

3.1. Chi-Square Measure for Term Co-Occurrence Pattern

For a term *w*, we are interested in measuring how unusual its co-occurrence pattern with frequent terms is. More specifically, a distribution of co-occurrence frequencies (with frequent terms) is converted to a chi-square value. Formally, for a term *w* in a document, the corresponding chi-square value is computed as [13],

$$\chi^2(w) = \sum_{g \in G} \frac{(freq(w,g) - n_w p_g)^2}{n_w p_g} \qquad (1)$$

where G denotes a set of frequent terms , n_w is the total number of terms in sentences (in G) where w appears, and p_g represents the expected occurrence probability of a frequent term g in G. In practice, p_g is computed as a fraction of the number of terms in all sentences where g appears divided by the total number of terms in the document. *freq(w,g)* is the co-occurrence frequency of w and g in the same sentences.

Two chi-square values are comparable only when their underlying frequency distributions have the same degrees of freedom. For a document, the corresponding chi-square distribution has the degrees of freedom of $|G|$ - 1, where $|G|$ denotes the number of frequent terms in the document. Clearly, the distribution would generally vary from one document to another. Consequently, chi-square values are only comparable when they are computed for terms from the same document.

To make two chi-square values computed from different documents comparable, we decided to compute the area under the distribution curve up-to the given chi-square value as a normalized score for a term. Specifically, let x_w be the chi-square value of term w computed from a distribution that has d degrees of freedom. Then the area under the distribution curve up-to x_w is computed as $p_w = P(x \le x_w) = F_d(x_w)$, where F_d denotes the cumulative distribution function of the chi-square distribution whose degrees of freedom is d. As the number of documents increases, computation of these probabilities (areas) for different documents becomes very costly. Thus it would be of interest to fastly approximate an area from a chi-square distribution of arbitrary degrees of freedom.

F_d is approximated by the standard normal distribution $\Phi(x)$ in several ways; see [25] for details. In particular, for large *d*,

$$F_d(x_w) \approx \Phi\left(\frac{x_w - d}{\sqrt{2d}}\right), \qquad (2)$$

A better approximation is given by

$$F_d(x_w) \approx \Phi\left(\sqrt{\frac{9d}{2}}\left\{\left(\frac{x_w}{d}\right)^{1/3} - 1 + \left(\frac{2}{9d}\right)\right\}\right), \tag{3}$$

Since we are interested in comparing two chi-square values from different degrees of freedom, we simply need to compute

$$z_w = \sqrt{\frac{9d}{2}}\left\{\left(\frac{x_w}{d}\right)^{1/3} - 1 + \left(\frac{2}{9d}\right)\right\}. \tag{4}$$

This is in fact an approximated z-transformation of a chi-square value. For documents with large d , we may use the first approximation as (4) is more computationally intensive. In this case, for each document we may pick

$$z_w = \frac{x_w - d}{\sqrt{2d}}. \tag{5}$$

The summary of the document ordering based on (5) is illustrated in Figure 1.

3.2. Robust Estimation of Weight Through Average Hierarchical Clustering

We note that chi-square estimation using (1) can be highly unreliable if the frequent term set G includes many terms that are semantically equivalent. Such semantically similar terms would better be merged to produce more reliable frequency distributions. More specifically, for a term w and a set of semantically similar frequent terms S, we compute the total co-occurrences of w with any frequent term in S. To identify sets of semantically similar frequent terms, we choose to apply hierarchical clustering over the initially found frequent terms. For the similarity measure between two frequent terms, we utilize Kullback-Leibler Distance, i.e. their co-occurrence distributions over the entire frequent terms are compared.

For a hierarchical clustering, determination of the cut-off value in the dendogram is a crucial, yet difficult task. Indeed our preliminary analysis revealed that the quality of the extracted keyphrases heavily depends on the carefully chosen cut-off values. To eliminate any spurious errors caused by improper choices, we averaged z-transformed chi-square values measured from different clustering results, where the number of clusters in each case is manually selected. Intuitively, keyphrases that consistently have higher z-transformed values with a large number of different clustering results should be given higher weights. Thus, averaging the values over different cluster groups eliminates sporadically appearing noisy keyphrases because of improper selection of clusters.

3.3. Document Ordering Using Vectors of *z*-Transformed Values

Our document ordering is based on the assumption that an initial list of keyphrases (with their *z*-transformed values) that denotes the topic of interest is available. Given such a vector of *z*-transformed values, $\boldsymbol{v} = (v_1, v_2, v_3, v_4, \dots, v_k)$, where v_i stands for the *z*-transformed value of the i^{th} keyphrase, the order between two documents is determined by computing their distances to $\boldsymbol{v}$ in a given metric space. More specifically, for document i, $\boldsymbol{x_i} = (x_{i1}, x_{i2}, \dots, x_{im})$, a vector of *z*-transformed chi-square values for keyphrases in i is computed using (1) and (4). Likewise, $\boldsymbol{x_j} = (x_{j1}, x_{j2}, \dots, x_{jm})$, is computed for document j. Then the order between i and j is determined by measuring distances between $\boldsymbol{v}$ and $\boldsymbol{x_i}$, and between $\boldsymbol{v}$ and $\boldsymbol{x_j}$; the smaller the distance, the higher the ranking. The overall ranking process is described in Figure 1.

```
Input: The topic of interest is given as $\boldsymbol{v} = (v_{i1}, v_{i2}, \dots, v_{im})$, which is a score vector for the keyphrases.
For each document $i$,
    For each keyphrase $j$
            1) Calculate $x_{ij}$ as in (1),
            2) Calculate $z_{ij}$ as in (4) (or (5) for documents of large degree of freedom),
    Calculate $s_i$, i.e. distance between $(x_{i1}, x_{i2}, \dots, x_{im})$ and $\boldsymbol{v}$.
For any pair of documents $i$ and $j$
    $s_i > s_j \Rightarrow r_i > r_j$, where $r_j$ denotes the ranking of document $j$.
```

Figure 1: The overall procedure of the document ordering. Keyphrase score vector ***v*** for the topic is assumed to be available.

4. Empirical Evaluation of Keyphrase Extraction Methods and Ranking

This section reports the results of an empirical study of the proposed document ordering framework using four sets of documents: 50 documents from the Department of Homeland Security's Information Analysis and Infrastructure Protection (DHS/IAIP), 6 documents from Aliweb, 6 journal papers and 8 documents from CSTR collection. DHS/IAIP is a daily report that summarizes open-source information regarding critical infrastructure issues. The Aliweb corpus is a collection of HTML web pages gathered by Turney through the Aliweb search engine for his study [8, 23]. CSTR is a collection of Computer Science

	10%	15%	20%	25%	30%	Avg
Precision	12	12.4	16.4	16	15.2	22.8
Recall	12.7	13.1	17.5	16.9	16.1	23.9
F-Score	12.31	12.71	16.88	16.4	15.6	23.28
Avg # Keys	0.6	0.62	0.82	0.8	0.76	1.14

Table 1. Performance of different clustering cut-offs when tested over a test set of 50 IAIP documents using Porter stemming. The last row indicates the average number of keywords identified.

Tech Reports which were included as part of the New Zealand Digital Library (http://www.nzdl.org).

Due to a practical difficulty in gathering objective assessment over a topic of interest, and thus the absence of the initial set of keyphrases (and their scores) that denotes the topic, we decided to indirectly assess the relevancy of the proposed framework in two ways. First we evaluated how the proposed scoring scheme identifies representative keyphrases from a document. This was conducted over 50 documents from DHS/IAIP, where a set of keyphrases is available for each document. In particular, the proposed scheme of averaging clustering results is closely examined. Second, the evaluation of the score of each keyphrase (and its *z*-transformed score) is performed manually by human evaluators using the other three document sets.

4.1. Averaging Clustering Results

First, we report the performance of average hierarchical clustering. To evaluate how averaged chi-square values help identify keyphrases, we manually set 5 different cut-off values in the dendogram. The cut-off values are determined in such a way that the number of clusters is 10%, 15%, 20%, 25%, and 30% compared to the total number of nodes in the dendogram. In other words, if we have 100 terms to be clustered, the number of clusters would be 10, 15, 20, 25 and 30. As clearly illustrated in Table 1, we achieve the best performance when chi-square values are averaged from all 5 clustering results.

Hierarchical clustering has different flavors when it comes to partitioning the data at each step of forming clusters. To study how each of these methods affects the keyphrase extraction, we tested complete, average, single, mcquitty, median, centroid and ward hierarchical clustering methods. The results of individual methods are summarized in Table 2. Mcquitty showed the best performance, whereas single linkage achieved the worst performance.

	Avg	Cent-roid	Com-plete	Mcquitty	Median	Single	Ward
Preci-sion	19.2	19.2	22.8	24	22.4	14.4	20.4
Recall	20.5	20.5	23.9	25.5	23.9	15.5	21.6
F-Score	19.77	19.77	23.28	24.67	23.06	14.89	20.93
Avg Keys	0.96	0.96	1.14	1.2	1.12	0.72	1.02

Table 2. Performance of Hierarchical clustering methods when tested over 50 IAIP document test set with Porter stemming.

4.2. Evaluation of Keyphrase Extraction

The performance of our corpus independent method was assessed by means of manual evaluation. In particular, we compared the performance of our co-occurrence based keyphrase extraction method with four other existing methods (see Table 3 for details). For each document, six sets of keyphrases are retrieved, one from the author-assigned list that came with the data set, and the other five from each method. We limit the number of keyphrases for each document to 15. In the case that a method produces *n* <15 keyphrases and it is the minimum of all methods, we select exactly *n* keyphrases from all other methods. Each document and its six keyphrase sets were presented to human evaluators. An evaluator was asked to assign a relevancy score to each keyphrase set. More specifically, within a scale of 1 to 10 (the higher the better), the evaluators are asked to:

- Evaluate how an individual keyphrase is relevant to the given document.
- Evaluate how the keyphrase set as a whole covers the topics in the document.

Then, five methods and author assigned keyphrases are ranked based on the scores given by the evaluators. Finally, the ranks are averaged over all the evaluators. This evaluation procedure is borrowed from the work of Jones and Paynter [26]. Whereas the Naïve Bayes corpus-dependent method shows the best (next to author assigned list) performance, our corpus independent method also demonstrates a competitive result (see Table 3).

4.3. Evaluation of *z*-Transformed Scores

The proposed document ranking method mainly depends on the relevancy of the transformed chi-square value. To evaluate how a *z*-transformed value

	Individual Keyphrase Quality			Topic Coverage		
	Avg	Std	Rank	Avg	Std	Rank
Author Assigned	5.8	1.7	10	5.9	1.2	7.4
Corpus-Dependent (Domain Specific)	4.9	1.2	9	6.6	0.6	9.4
Corpus Dependent (Domain Unspecific)	4.7	1.3	7.8	6.4	0.7	8.4
TF-IDF	4.6	1.3	6.9	5.9	1.2	7.4
TF	4.1	1.5	5.4	5.2	1.1	5.2
Corpus-Independent	4.5	1.4	6.8	5.8	1.3	7.4

Table 3. Results based on human evaluation of key phrases extracted from 20 documents. Std denotes the standard deviation measured from 20 documents.

represents a degree of relevancy, we extracted 15 keyphrases from 20 documents (6 documents from Aliweb, 6 journal papers and 8 documents from CSTR collection) based on local chi-square values x_i. Each keyphrase was then assessed based on its degree of relevancy by human evaluators. Finally chi-square values of all keyphrases were transformed to z-scale following (3) for comparison purpose. Note that we again averaged five z-transformed scores obtained from different levels at the dendogram. Table 4 illustrates the correlation between z-transformed values and their assessment from human evaluators. Although the experiment is not a large-scale and is preliminary, it clearly demonstrates that a keyphrase with high z-transformed value tends to be assessed more relevant, which essentially suggests that the proposed document ranking method is promising.

5. Conclusions and Future Direction

In this paper, we proposed a novel method that maintains rankings among documents within a domain of interest. Unlike conventional approaches, our method does not require a large corpus of training documents, nor does it require re-assessment of existing documents upon receiving a new collection. As demonstrated in our empirical study, the proposed method successfully identifies relevant keyphrases solely by comparing z-score values. Considering the importance of document ordering and the number of documents newly available each day, our corpus-independent document ordering method will likely benefit many information processing infrastructures, and also find numerous practical applications.

Our future work aims to refine the method in several directions. For example, a more robust strategy for computing chi-square values is anticipated. Specifically, instead of manually determining cut-points in a dendogram, we need to identify optimal cut-points using more elaborate methods like model-based hierarchical clustering algorithms. Indeed`, we are currently investigating various methods in this direction including classification likelihood based hierarchical agglomeration method using EM [24].

z-score Range	0.5 ~ 0.69	0.7 ~ 0.79	0.8 ~ 0.99
Average Ranking	4.28	5.64	5.87

Table 4. Average rankings of keyphrases at different z-score ranges. Keyphrases are sorted to their z-score and average ranking are measured at different ranges.

References

[1] G. K. Zipf, Human Behavior and the Principle of Least Effort. Cambridge, Mass.: Addison-Wesley, 1949.

[2] W. B. Croft and D. J. Harper, "Using probabilistic models of document retrieval without relevance information " in Document retrieval systems, Taylor Graham Series in Foundations of Information Science, pp. 161-171, 1988.

[3] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," Journal of Documentation, vol. 28, pp. 11-21, 1972.

[4] D. Harman, "Ranking Algorithms," Information Retrieval: Data Structures & Algorithms (Ed.: Frakes & Baeza-Yates), 1992.

[5] G. Salton and C. Buckley, "Term Weighting Approaches in Automatic Text Retrieval," Information Processing and Management, vol. 24, pp. 513-523, 1998.

[6] S. E. Robertson, S. Walter, M. Hancock-Beaulieu, A. Gull, and M. Lau, "Okapi at TREC-4," proceedings of the 4th Text REtrieval Conference (TREC-4), Gaithersburg, Maryland, November 1995.

[7] P. D. Turney, "Mining the Web for synonyms: PMI-IR versus LSA on TOEFL," proceedings of the 12th European Conference on Machine Learning (ECML-2001), Freiburg, Germany, September 2001.

[8] P. D. Turney, "Learning Algorithms for Keyphrase Extraction," Information Retrieval, vol. 2, pp. 303-336, 2000.

[9] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning, "KEA: Practical automatic keyphrase extraction," proceedings of the 4th ACM Conference on Digital Libraries, Berkeley, California, August 1999.

[10] P. D. Turney, "Coherent Keyphrase Extraction via Web Mining," proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, August 2003.

[11] K. W. Church and P. Hanks, "Word Association Norms, Mutual Information, and Lexicography," Computational Linguistics, Vol. 16, No. 1, pp. 22-29, 1990.

[12] C. D. Manning and H. Schutze, Foundations of Statistical Natural Language Processing. Cambridge, Massachusetts: MIT Press, 1999.

[13] Y. Matsuo and M. Ishizuka, "Keyword extraction from a single document using word co-occurrence statistical information," International Journal on Artificial Intelligence Tools, vol. 13, pp. 157-169, 2004.

[14] Y. Ohsawa, N. E. Benson, and M. Yachida, "KeyGraph: Automatic Indexing by co-occurrence graph based on building construction metaphor," proceedings of the IEEE International Forum on Research and Technology Advances in Digital Libraries (ADL '98), Santa Barbara, California, April 1998.

[15] A. Trotman, "Learning to rank," Information Retrieval, vol. 8, pp. 359-381, 2005.

[16] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," Journal of the ACM, vol. 46, pp. 604-632, 1999.

[17] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," Computer Networks and ISDN Systems, vol. 30, pp. 1-7, 1998.

[18] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP), Barcelona, Spain, July 2004.

[19] G. Erkan and D. R. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization," Journal of Artificial Intelligent Research, vol. 22, pp. 457-479, 2004.

[20] I. Dagan, L. Lee, and F. C. N. Pereira, "Similarity-based models of word co-occurrence probabilities," Machine Learning, vol. 34, pp. 43-69, 1999.

[21] F. Pereira, N. Tishby, and L. Lee, "Distributional clustering of English words," proceedings of the 31st Meeting of the Association for Computational Linguistics (ACL), Colombus, Ohio, pp. 183-190, June 1993.

[22] E. B. Wilson and M. M. Hilferty, "The distribution of chi-square," proceedings of the National Academy of Sciences of the United States of America, vol. 17, pp. 684-688, 1931.

[23] P. D. Turney, "Learning to extract keyphrases from text," Technical Report ERB-1057, National Research Council, Institute for Information Technology, February 1999.

[24] C. Fraley and A. E. Raftery, "Model-Based Clustering, Discriminant Analysis, and Density Estimation," Journal of American Statistical Association, vol. 97, pp. 611-631, 2002.

[25] N. L. Johnson, S. Kotz, and N. Balakrishnan, “Continuous Univariate Distributions” vol. 1, Second Edition, John Wiley & Sons, Inc., 1994.

[26] S. Jones, and G. W. Paynter, “An Evaluation of Documents Keyphrase Sets”, Journal of Digital Information, vol. 4 Issue 1, 2003.