Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 73 (2014) 26199-26204

ISSN: 2229-712X

An Improvement to TF*PDF: Salient Long Running Event Detection from News Documents based on various features

Y. Jahnavi and Y. Radhika

Computer Science & Engineering Department, GITAM University, Visakhapatnam, Andhra Pradesh-517 502, India.

ARTICLE INFO

Article history:

Received: 21 August 2013;

Received in revised form:

25 July 2014;

Accepted: 5 August 2014;

Keywords

Long running events,

TDT,

Term Weighing,

TF-IDF,

TF*PDF,

Similarity Measures,

Evaluation Measures.

ABSTRACT

Automated extraction of popular news from the freely accessible news corpora is becoming important in today's internet world. Because of the availability of large volumes of news wire sources, it is hectic for the human beings to search and decide whether it is popular or not. This necessitates a tool which should extract hot news in a period of time. Term weighting is a useful technique which extracts salient features from the text documents. Though, there exist different tools based on different term weighting algorithms, these are inaccurate in the extraction of hot news. In this paper, a new feature extraction algorithm for long running events based on frequency, position, scattering and topicality is proposed. Experimentation has been done on different retrospective news wire sources. Experimental results demonstrate that the proposed algorithm is suitable for extracting hot news.

Introduction

Data Mining techniques have been developed in order to extract knowledge from raw data repositories. Various data mining techniques such as Frequent Pattern Mining, Association Rule Mining, Classification, Clustering, Outlier Analysis etc., have been developed to extract knowledge which is useful for many real time applications. Data may be of any type such as Relational, Transactional, Spatial, Text, Multimedia etc. Various approaches handling with unstructured documents are Information Retrieval, Information Extraction, and Text Mining. Information Retrieval Systems are useful for extracting useful documents from the large collections of stored documents [1]. Unlike information retrieval which concerns how to recognize relevant documents from a document collection, information extraction generates structured data organized for further processing [2]. Text mining research area is useful for finding patterns in text collections. These patterns are the extraction of topics from texts or grouping of text or the identification of trends [3].

The crucial steps in the IR process are the document representation and query representation [4]. In the document representation, the documents should be pre-processed and the terms in each document are weighted. To expedite the accessing, these documents are indexed. The queries are represented by means of Boolean logic, proximity, continuous word phrases, fuzzy searches, term masking, ranking, canned queries etc [1]. These queries are compared with the documents by using different techniques wherein exact match the system finds the documents that fill all the conditions of a Boolean query (it predicts relevance as 1 or 0). To increase recall, the system can exploit synonym expansion and hierarchic expansion [5]. To improve the accuracy of IRS, relevance feedback was introduced by Rocchio in 1965. In this mechanism, the search process often goes through numerous iterations: knowledge of the features that distinguish relevant from irrelevant documents is used to improve the query or the indexing. The new query should be based on the old query modified to enhance the weight of terms in relevant items and lessen the weight of terms that are in non-relevant items [1].The system matches the stored documents with the query and displays the documents that are similar to the query. The user selects relevant items based on their interest. The gauge of IRS is done by different evaluation measures such as precision, recall, F-measure etc [6].

Information Extraction produces templates from texts. To automate the conversion of input pages into structured data, numerous efforts have been stanch in the area of information extraction. Unlike information retrieval which concerns how to identify relevant documents from a document collection, information extraction produces structured data prepared for post processing, which is vital to numerous applications [7].

Text mining research area is useful for finding patterns in texts. These patterns are the extraction of topics from texts or grouping of text or the identification of trends [8]. Categorization of documents can be done manually or automatically. In manual categorization, the identification of topics should be done by the subject experts. But manual categorization is expensive; therefore the research has focussed on automatic categorization. There are two types of automatic categorization, supervised categorization (classification) and unsupervised categorization (clustering). Since we don't know the name of the classes in prior, the proposed algorithm is under clustering [9].

NLP is the endeavour to extract a fuller significance representation from free text. NLP imposes certain linguistic rules that extract a fuller meaning representation from free text. It naturally makes use of linguistic concepts such as Parts Of Speech tagging (Noun, Verb, and Adjective etc) and grammatical structure (either represented as phrase like noun phrase or prepositional phrase, or dependency relations like subject- of or object -of). NLP is started in the year of 1960 which is used for studying cohort and understanding of natural languages. There are different applications of NLP such as

Tele:

E-mail addresses: yjahnavi.2011@gmail.com

Information Retrieval, Information Extraction, Text Mining (Text grouping, Question-Answering, Language Translation, Opinion mining, Text Summarization etc.) [10]. There exists different functionalities of NLP such as Morphological and lexical processing, Syntactic Analysis, Semantic Analysis etc.

Most of the Text based techniques use vector space models in which the documents and the queries can be represented as vectors, which can be used to search their nearest neighbours in a document collection. However, for any nontrivial document database, the number of terms T and the number of documents D are usually relatively large. Such dimensionality leads to the difficulty of inefficient computation, since the consequential frequency table will have size T X D. Furthermore, the high dimensionality also leads to very sparse vectors and raises the difficulty in detecting and developing the relationships among them. To overcome these problems, dimensionality reduction techniques such as Latent Semantic Indexing, Probabilistic Latent Semantic Indexing and Locality Preserving indexing can be used [17].

Different sources of text are news papers, books, journals, magazines, manuals, emails, blogs etc [11]. Topic detection and tracking (TDT) is a research inventiveness concerned with the techniques to organize news documents. In contrast to the more traditional information retrieval problems, the focus in TDT is on news events: In breaking the text into cohesive stories, spotting something previously unreported, tracking the progress of the event, and grouping together news that discuss the same event. This area has also been called event-based information organization.

Article Representation

There are several ways to represent a text document and articles in the text document. The most popular representation in Information Retrieval and Text Mining is Vector Space Model, which is also called as Bag of Words approach. It represents each article as a feature vector of terms where each feature vector contains term weights or term frequencies. Although VSM doesn't represent the relationship between terms, it is more popular because of its mature theories and efficient computational performances.

The Vectors are represented as

	Word 1	Word 2	.....	Word n
A1	w_{11}	w_{12}	.....	w_{1n}
A2	w_{21}	w_{22}	.....	w_{2n}
.....	.....	.....	.....	.....
Am	w_{m1}	w_{m2}	.....	w_{mn}

People have hypothesis that the Phase based approaches could perform better than the term based ones. Although phases are less ambiguous, they have low frequency of occurrence and the data may contain large number of noisy phases. Various term weighting algorithms are used for finding the weight of each term in the vector. Term frequency component may be a binary weight (i.e., 1 if the term exists in the document or 0 if the term doesn't exist) or raw term frequency (i.e., the number of occurrences of the term in document) or normalized term frequency. To increase the retrieval performance collection frequency has been introduced. The most popular collection frequency component is inverse document frequency (idf). To equalize the length of the document vectors, normalization component has been introduced [12].

Later different variations of term weighting algorithms have been proposed. Term weighting algorithms are broadly classified into supervised and unsupervised where supervised term weighting algorithms use known membership information and unsupervised algorithms do not use such information. TF-IDF weights are calculated from the mixed corpus, whereas TF*PDF weights are calculated from each channel and these channel wise weights are summed up. Later different variations of TF*PDF also have been proposed [13][14].

Basically, there exist two types of news events such as short lived events and long running events. The popularity of the short lived events elapses in a few time slots, whereas the long running events live for a period of days. As the media and people continuously focus on a particular topic for a long period, the popularity of such event lies in a large time span even as time goes on. It is well known fact that a news text stream is a sequentially ordered documents, to extract long running events the topic extraction model should not only focus on term's frequency, but also on various statistical features of terms. In several papers it is found that only a few researchers have considered topicality which is an evident feature that affects the hot term extraction process. So far a large research has evolved for the extraction of short term events. The main objective of this paper is to consider various statistical term features for extracting long running events from retrospective news corpora. Experimental confirmations were carried out to identify the effectiveness of the proposed method.

By using χ^2 statistic, Swan and Allan proposed a model to automatically identify topics for the timeline from the set of news corpus [15]. Chen proposed an event life cycle model based on Aging Theory for news events. The popularity of news event is not fixed overtime i.e., it is in different life forms such as birth, growth, decay and death [16].

Salient Long Running Event Detection

The proposed method extracts seminal hot terms by considering the new weight as the parametric summation of Frequency, Position, Scattering and Topicality within a Time Span. Time Span which essentially corresponds to a period of days in the news text stream sequence, it is defined as TimeSpan = [ss, es], where ss represents the start time of this time window, es records the end of the time window. A time span is a range of time slots where a slot is the representation of a day [15]. Channel wise, Time Sliced text documents are the input to the proposed model. Each sentence in the document has well defined boundaries.

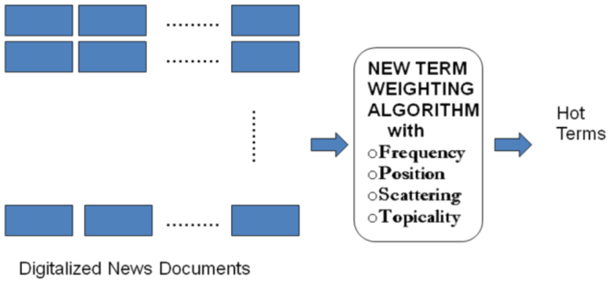

Fig.1: System design

Preprocessing is the essential step in Information Retrieval Systems and Text Mining. Since it is not possible to perform all operations on all the terms in each news document, preprocessing should be performed to extract a short list of terms. First of all, the documents from all the news wires should be stored channel wise and time slot wise. Then Tokenization is performed to build sentence boundaries. All the stop words should be removed from all the documents. And then Porter's stemmer algorithm [18] is used to extract the stemmed list of terms. Stemming plays an important role that extracts the stems from similar meaning content words. Frequencies are counted for all the terms in the documents other than for the terms in the title, a rank is then established and the terms above a specified threshold are selected for hot term extraction list.

As the presence of the term increases in more number of documents in most of the channels within the time span (Frequency), the popularity of the term is more. Channel wise frequency of each term is calculated by the existing TF*PDF algorithm. This algorithm indicates that the weight of a term in each channel is linearly proportional to the normalized term frequency and inversely proportional to the ration of the documents containing the term to the total number of documents in that channel [19].

$$W_j = \sum_{c=1}^{c=D} |F_{jc}| \exp\left(n_{jc}/N_c\right) \quad (1)$$

Where

W_j =Weight of term j;

Normalized term frequency $|F_{jc}| = \frac{F_{jc}}{\sqrt{\sum_{k=1}^{k=K} F_{kc}^2}}$

F_{jc} =Frequency of term j in channel c;

n_{jc} =Number of documents in channel c, where term j occurs;

N_c =Total number of documents in channel c;

K =Total number of terms in a channel;

D =number of channels.

Frequency is the most important factor, but this factor alone cannot extract seminal terms. For extracting hot terms, Position of the term in the news document should be considered. The terms which appear in the title of the news document should be weighted more. More articles may discuss about the same term, if it is more important topic. Thus that term may appear in more number of titles. Therefore the position weight of a term is simply one if it appears in the title or zero if not.

The Position weight of term j is

$$PW_j = \begin{cases} 1 & j\ appears\ in\ the\ title \\ 0 & j\ doesnot\ appear\ in\ the\ title \end{cases} \quad (2)$$

The term's weight should be still more improved if it appears in more number of articles.

Scattering of each term is calculated as

$$S(j) = \sum_{s=ss}^{es} \sum_{d=1}^{|D|} \frac{a}{A} \quad if \quad PW_j > 0 \quad (3)$$

$S(j)$ =Scattering of term j;

a= Number of articles in which this term t appears in the current document;

A= Total number of articles in the current document in the current time slot in the time span;

$|D|$ =Total number of documents in each time slot in the time span.

According to the model proposed by chen [13], the later property is calculated by tracking the life cycle of the term. According to the life cycle model, the energy $E_{j,s}$ of term j measures the frequency of term j appearing in a specified time slot s by comparing with other slots.

The energy of term j on channel c and time slot s is calculated by using χ^2 test [15],

$$E_{c,s}(j) = \chi^2 = \frac{(A+B+C+D)*(AD-BC)^2}{(A+B)(A+C)(B+C)(B+D)} \quad (4)$$

Where A: The number of occurrences of term j on the current channel in the current time slot;

B: The number of occurrences of term j on other channels in the current time slot;

C: The number of occurrences of term j on the current channel in other time slots;

D: The number of occurrences of term j on other channels in other time slots.

Energy of term j in time slot s from all the channels is calculated as

$$E_{js} = \sum_{c \in C} E_{c,s}(j) \quad (5)$$

The accumulated energy AE_{js} of each term j at each time slot s in the time span is also calculated. The life support value of term j at time slot s is calculated as the naturallogarithm of accumulated energy AE_{js} represented as lifesupport$_{j,s}$. Term's energy is reduced by the energy decay which is an empirical constant.

Topicality for long running events is the mean of life supports in each time slot for each term, which can be computed as:

$$T_j = \frac{\sum_{s=ss}^{es} life\sup port_{j,s}}{N} \quad (6)$$

Where $life\sup port_{j,s}$ is the lifesupport of term j in time slot s;

N is the number of time slots in the selected Time Span.

The overall weight of term t is measured by the parametric summation of all these properties.

$$T_{new} = \alpha_1 W_j + \alpha_2 PW_j + \alpha_3 S(j) + \alpha_4 T_j \quad (7)$$

The values of α_1, α_2, α_3 and α_4 are adjustable parameters.

These parameters are useful for giving varying importance to different weights.

Hot articles are those which contain more number of hot terms. After finding term weights T $_{new}$ for all terms, we extract hot terms whose weight is greater than the specified threshold. In the next step, hot articles are extracted with these hot terms. The weight of the hot article is the average weight of all the term that it contains and clustering of articles is performed which gives hot topics.

Similarity Measures

The similarity measures reflect the degree of closeness between different objects and thus groups the most similar objects into the clusters.

Euclidean Distance Measure

It is the most commonly used method for finding distance between two samples. It is used for the data sets which are normalized and is a default measure for k-means [1] [21]. It is a standard metric for geometrical problems which calculates the difference between two samples directly based on the magnitude.

$$E\left(\vec{t}_a, \vec{t}_b\right) = \sqrt{\left(\sum_{t=1}^{m} \left|w_{t,a} - w_{t,b}\right|^2\right)} \quad (8)$$

Where

E is the Euclidean distance between two term vectors;

$\vec{t}_a$ = Term vector of document d_a;

$\vec{t}_b$ = Term vector of document d_b;

m = Number of terms;

$w_{t,a}$ and $w_{t,b}$ are term weights.

Cosine Similarity

If the documents are represented in term vectors, then the similarity of the two documents is determined by the correlation between the vectors. It is the popular similarity measure for text documents which is defined as the cosine of the angle between vectors [1] [21]. In this measure the important property is that the documents with same composition but different totals will be treated identically [1].

$$SIM(Doc_i, Doc_j) = \frac{\sum_{t=1}^{m}\left(Doc_{it} \times Doc_{jt}\right)}{\sqrt{\sum_{t=1}^{m}\left(Doc_{it}\right)^2} \times \sqrt{\sum_{t=1}^{m}\left(Doc_{jt}\right)^2}} \qquad (9)$$

Doc_{it} is the tth term in the document vector i;

Doc_{jt} is the tth term in the document vector j;

m total number of terms in that document.

Cosine similarity is bounded between [0, 1]. It is independent of document length. If the value is one then the two documents are identical and if it is zero then there is nothing common between them.

Jaccard Coefficient

In the Jaccard similarity measure, the denominator becomes dependent upon the number of terms in common. As the common elements increase, the similarity value rapidly decreases, but is always in the range -1 to +1. It is sometimes referred as Tanimoto coefficient. Since this measure depends on common terms, the number of common terms in the vector increases then this value becomes negative [21].

$$SIM(Doc_i, Doc_j) = \frac{\sum_{t=1}^{m}\left(Doc_{it} \times Doc_{jt}\right)}{\sum_{t=1}^{m}\left(Doc_{it}\right) + \sum_{t=1}^{m}\left(Doc_{jt}\right) - \sum_{t=1}^{m}\left(Doc_{it} \times Doc_{jt}\right)} \qquad (10)$$

Doc_{it} is the tth term in the document vector i;

Doc_{jt} is the tth term in the document vector j;

m total number of terms in that document.

Dice Coefficient

This measure simplifies the denominator of Jaccard coefficient and introduces a factor 2 in the numerator. This measure doesn't depend on the common terms. As long as the vector values are same, this measure normalization factor remains unchanged [1].

$$SIM(Doc_i, Doc_j) = \frac{2 \times \sum_{t=1}^{m}\left(Doc_{it} \times Doc_{jt}\right)}{\sum_{t=1}^{m}\left(Doc_{it}\right) + \sum_{t=1}^{m}\left(Doc_{jt}\right)} \qquad (11)$$

Doc_{it} is the tth term in the document vector i;

Doc_{jt} is the tth term in the document vector j;

m total number of terms in that document.

In [22], the authors have proved that the cosine measure is superior to other measures such as Jaccard and Euclidean measure. We have used the cosine similarity measure for finding the similarity between different articles.

Experimental Setup

Evaluation of the proposed algorithm has been done by collecting text articles from Hindu, Indian Express and Times of India news articles dated from 11th August 2011 to 11th September 2011. Channel wise and Times Slot wise stored text articles are input to the proposed Term weight based Hot Miner. After pre-processing (segmentation, stop word removal, stemming, pruning) all the documents, the proposed Term weighting algorithm has been performed to extract hot terms. All the terms extracted by the proposed algorithm are heavily weighted because it considers more features and most of the extracted terms are Named Entities. Table.1 shows the extracted hot terms by the existing TF*PDF algorithm and the proposed Term weighting algorithm.

SNO	Terms extracted by TF*PDF	Weight	Terms extracted by the proposed FPST Algorithm	Weight
1	Hazar	1.2089	Hazar	2.3269
2	Anna	0.8616	Polic	2.060
3	Minist	0.4962	Anna	1.9766
4	Fast	0.4919	Protest	1.8758
5	Support	0.4645	Govern	1.8204
6	Singh	0.4492	Delhi	1.7751
7	Arrest	0.4124	Corrupt	1.6709
8	People	0.3612	Park	1.6602
9	Dai	0.3594	Minist	1.6065
10	Bill	0.3315	Fast	1.6013
11	Parti	0.3311	Support	1.5737
12	Team	0.3215	Arrest	1.538
13	Today	0.3115	People	1.4655
14	Tuesdai	0.2847	team	1.4473
15	Anti	0.2612	lokpal	1.4335
16	Court	0.2364	Bill	1.4332
17	Congress	0.2312	anti	1.3864
18	Year	0.1733	court	1.3556
19	Mondai	0.1729	detain	1.3395
20	Larg	0.1724	congress	1.3293
21	Nation	0.1723	ramalila	1.3344
22	Include	0.1695	state	1.328
23	Senior	0.1501	govt	1.3108
24	Number	0.1496	activist	1.2945
25	Visa	0.1424	Cbi	1.2961
26	Met	0.1421	station	1.2878
27	Back	0.1405	india	1.2855
28	Leader	0.1258	kejriw	1.2328
29	Arrest	0.1221	janmastami	1.2022
30	Activist	0.1158	manmohan	1.2021

Table 1: Hot Terms extracted by TF*PDF vs Proposed algorithm

Evaluation

Evaluation measures are used to evaluate the performance of clustering algorithm. There exists a wide variety of clustering measures based on the properties of homogeneity and completeness, among them the most popular evaluation measures are described in this paper.

Entropy

Entropy is a commonly used measure in information theory for finding the quality of cluster. Entropy can be considered as a measure of uncertainty which evaluates the distribution of categories in a given cluster [23]. Entropy reflects the quality of individual clusters in terms of homogeneity of data points in the cluster.

$$E\left(C_i\right) = -\frac{1}{\log c}\sum_{h=1}^{k}\frac{n_i^h}{n_i}\log\left(\frac{n_i^h}{n_i}\right) \qquad (12)$$

C = Total no. of categories in the data set;

h = The number of documents from h class;

n_i

n_i = Numbers of documents belonging to cluster i;

i varies from 1 to k;

k total number of classes or clusters.

Equation (12) represents class entropy. The overall class entropy is called as cluster entropy and it is defined as,

$$entropy = \sum_{i=1}^{k} \frac{n_i}{n} E(C_i) \qquad (13)$$

n_i = Numbers of documents belonging to cluster i;

n = Total number of clusters.

Both the cluster entropy and class entropy utilizes the predefined class labels on input data. These measures are practically capable of evaluating any clustering system.

Purity

To compute this measure, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by 'n' [23]. If each document gets its own cluster then purity is 1.

$$purity(n,c) = \frac{1}{n}\sum_{k=1}^{n} \max_j \left| w_k \bigcap c_j \right| \qquad (14)$$

n = Number of documents;

c_j = Set of documents in cluster j;

w_k = Set of documents in class k;

It is a simple and transparent evaluation method.

Precision

It is also called as reproducibility or repeatability which measures the consistency of the results. It is related to sampling error which is the difference between the sample eliminate and the population value [20].

It is the degree to which the repeated measurements under unchanged condition show same results. The precision increases with the size of the sample.

In field of information retrieval, it is the fraction of retrieved documents that are relevant to the retrieved documents.

$$precision = \frac{\left|\{relavantdocuments\} \bigcap \{retrieveddocuments\}\right|}{\left|\{retrieveddocuments\}\right|} \qquad (15)$$

It takes all the retrieved documents into account, but it can also evaluate at given cut-off.

In binary classification, $precision = \frac{t_p}{t_p + f_p}$

Where t_p true positive;

f_p false positive.

Recall

It is the fraction of the documents that are relevant to the query which are successfully retrieved [10].

In binary classification, $recall = \frac{t_p}{t_p + f_n}$

Where t_p true positive;

f_p false negative.

It is called as sensitivity.

Recall alone is not enough to measure the number of non relevant documents.

F-Measure

It is used to balance the contribution of all negatives. It is the harmonic mean of precision and recall values used in information retrieval [20].

$$FScore = \sum_{i=1}^{k} \frac{n_i}{n} \max_j \left(F_{i,j}\right) \qquad (17)$$

$$where\, F_{i,j} = \frac{2 \times P_{i,j} \times R_{i,j}}{P_{i,j} + R_{i,j}}$$

$$p_{ij} = \frac{n_{ij}}{n_j}; R_{ij} = \frac{n_{ij}}{n_i}$$

n_{ij} =Number of documents in class i and cluster j;

n_i =Number of documents in class i;

n_j =Number of documents in cluster j.

Coverage Rate

It is used to find the percentage of extracted topics to the actual topics [20].

$$Coverage \quad Rate = \frac{Extracted\,Topics}{Actual\,Topocs} \times 100\% \qquad (18)$$

To use this measure, manual categorization should be done for calculating the percentage of documents in each topic and then the ratio is calculated.

We manually grouped the collected news articles into distinct clusters and checked the extracted topics with the manual clusters. The results represented in Table 2 shows that the proposed algorithm has more coverage rate. i.e., more number of hot articles is extracted from the total number of actual hot topics.

Table 2: Comparison of existing TF*PDF with the Proposed Algorithm

Number of Articles	TF*PDF	New Algorithm
150	30.76%	38.46%
300	46.15%	53.84%
600	61.53%	69.23%
750	69.23%	76.92%
900	84.61%	92.30%

Conclusion and Future Scope

The following conclusions are drawn from our experimental results. Text clustering is a research focus and important subtask in the field of Text Mining. By considering various features an improved Term Weighting algorithm for handling text documents in the real world is proposed. Newly proposed term weighting method achieves the best performance regularly and outperforms other methods substantially and considerably. The proposed algorithm doesn't consider the semantic relationship between the terms. Future research may consider natural language processing techniques to establish the relationship between the terms.

References

[1] J. Kowalski and Mark.T. Maybury, "Information Storage and Retrieval Systems Theory and Implementaion", 2nd Edition, Springer publications.

[2] Chia-Hui Chang, Mohammed Kayed, Moheb Ramzy Girgis, Khaled Shaalan," A Survey of Web Information Extraction Systems", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, TKDE-0475-1104.R3.

[3] S. M. Indurkhya, N. Zhang, T. Damerau, F. Weiss, "Text Mining Predictive Methods for Analyzing Unstructured Information," Springer, 2005.

[4] 8. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, "An Introduction to Information Retrieval", Cambridge University Press, April 1, 2009.

[5] 7. Amit Singhal ,"Modern Information Retrieval: A Brief Overview" , IEEE,2001.

[6] R Jizba, "Measuring Search Effectiveness", creghton.edu, 2007.

[7] Anna Huang department of computer science, "Similarity measures for text documents clustering", university of Waikato , Hamilton , Newzealand 2004.

[8] A. Kao, S. Potect, "Text Mining and Natural Language Processing Introduction for the Special Issue", SiGKDD Explorations, 2004, vol.7, Issue, pp.1-3.

[9] Wang, F., C. Zhang and T. Li, "Regularized clustering for documents", Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, Netherlands, July 2007, 23-27, , pp:95-102.

[10] Daniel Jurafsky, James H. Martin, "Speech and Language Processing", An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, University of Colorado, Boulder.

[11] Guoliang Shi, Yangqing Kong, "Advances in Theories and Applications of Text Mining", ICISE, 2009, pp. 4167-4170.

[12] G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval", Information Processing and Management: An Int'l J., 1988, vol. 24, no. 5, pp. 513-523.

[13] Chen, Luesak Luesukprasert, and Seng-cho T. Chou, "Hot Topic Extraction Based on Timeline Analysis and Multidimensional Sentence Modeling", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2007,VOL. 19, NO. 8, pp. 1016-1025.

[14] Y. G. J. Liu, P. Ma, he, "Hot Keyphrase Extraction based on TF*PDF", International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11, 2011, pp. 1524-1528.

[15] R. Swan and J. Allan, "Extracting Significant Time Varying Features from Text", Proc. Eighth Int'l Conf. Information and Knowledge Management (CIKM '99), 1999, pp. 38-45.

[16] C. C. Chen, Y.T. Chen, Y. Sun, and M.C. Chen, "Life Cycle Modeling of News Events Using Aging Theory," Proc. 14th European Conf. Machine Learning (ECML '03), pp. 47-59, 2003.

[17] Scott Deerwester, Susan T. Dumais, Richard Harshman, "Indexing by Latent Semantic Analysis".

[18] M. Porter, An Algorithm for Suffix Stripping, Program, vol. 14, no. 3, 1980.

[19] Bun and M. Ishizuka, "Topic Extraction from News Archive Using TF_PDF Algorithm," Proc. Third Int'l Conf. Web Information Systems Eng. (WISE '02), 2002, pp. 73-82.

[20] R Jizba, "Measuring Search Effectiveness", creghton.edu, 2007

[21] Anna Huang department of computer science, "Similarity measures for text documents clustering", university of Waikato , Hamilton , Newzealand 2004.

[22] R. Subhashini and V. J. Senthil ," Evaluating the Performance of Similarity Measures used in Document Clustering and Information Retrieval", First International Conference on Integrated Intelligent Computing, IEEE, 2010.

[23] H. Kremer, P. Kranen, T. Jansen, T. Seidl, "An Effective evaluation measure for clustering on evolving data streams", 2011.