International Journal of Multimedia and Ubiquitous Engineering
Vol. 9, No. 12 (2014), pp. 97-106
http://dx.doi.org/10.14257/ijmue.2014.9.12.09

On Objective Keywords Extraction: Tf-Idf based Forward Words Pruning Algorithm for Keywords Extraction on YouTube

Ambele Robert Mtafya, Dongjun Huang and Gaudence Uwamahoro

Central South University, Changsha, Hunan, China;
Dar es salaam Institute of Technology, Tanzania.
kakaambe@gmail.com

Abstract

Discovery and subsequent effective retrieval of useful user generated content depends on proper meta-data annotation implemented on an object such as a title and Keywords. In this study, a simpler unsupervised non graph-based algorithm for extracting keywords is proposed. A novel key phrases chunking approach was adopted; this utilizes words sequences as they appear in the original document. The simple but effective Term frequency-inverse document frequency (tf-idf) weighting scheme was exploited to rank the novelty created key-phrases. Comparing to a similar algorithm that uses three metrics weighting scheme, the tf-idf yielded a precision of 89%.Thus, the application of tf-idf algorithm on YouTube's metadata based keywords shows to be useful approach in its objectivity.

Keywords: *keywords automatic extraction, Tf-Idf Weighting, Forward Words Pruning, Objective keywords, User generated content*

1. Introduction

As the results of web 2.0 technological revolution, the web is filled up with the huge amount of both professional and amateur user generated content (UGC). This digital resource is composed of wide variety of contents, with different formats and metadata types constitute a heterogeneous set of resources difficult to deal with [1]. UGC gained their importance through social media tools that facilitates online social networking and allow users to create and consume digital content at the same time via variety of channels .Discovery and subsequently effectively retrieving of UGC in social media domain is a continuous area of research interest. Some UGC in content sharing sites has a great educational potential; In YouTube's video collection for instance there is a large amount of user generated content that can be used for formal and informal learning [2]. The well- established methods for such digital resources discovery are within the text-based information retrieval domain which capitalizes on the document metadata, including tags and keywords extraction. Social media tagging is a compromise between manual and automatic indexing [3]. It is done by diversity of users who may or may not be experts consequently user generated tags varies widely. Another challenge with UGC tagging is that they directly reflect the conceptual and linguistic structure of the users and their diverse geographical and cultural backgrounds [4], this is prone to introduce biasness and semantic variation of the same content [1]. Thus, it is necessary to have ongoing studies that address these issues. In this study we propose a keyword extraction method that objectively creates keywords from various UGC tools like YouTube video, basing solely on the Content's metadata. The automatic annotation of UGC might be helpful to the content creators as well as to those who seek the content for various uses.

ISSN: 1975-0080 IJMUE

2. Related Work

2.1. Machine Learning Approaches

Document oriented keywords extraction are the preferred method over corpus oriented because of its reliability. The document based methods are not pertubated by changes like it is with corpora [5]. Within Document based approaches there are also supervised or unsupervised options; the most powerful key phrases extraction algorithms are based on supervised learning which address the problem of associating keyphrases to documents as a classification task. However, since this approach requires a corpus of similar documents, which is not always readily available therefore constitutes a major drawback. Therefore, even though The unsupervised approaches are, less accurate they are favored because they don t bring the massive training overhead while at the same time delivering fairy acceptable results for many applications as compared to their counterparts. Work for keywords or keyphrases extraction has been performed on the written text domain, often based on frequency, word association, structure or position, and linguistic knowledge.

2.2. Key Techniques

Some of the studies in unsupervised category includes the comparative algorithm combination by Zede Zhu, *et al.*, [6], typical graphical measure based by Lahiri, *et al.*, [7], Vocabulary expansion approach by Liu, *et al.*, [8], the nearest neighbor approach by wan and Xiao [8], and the classical TextRank and a graphical approach by Mihalcea and Tarau [5, 9, 10]. Regardless of the previous machine learning classification, and keyword extraction there are other interesting key techniques that worth mentioning , these include a Multi-relational Network Construction by Lei, *et al.*, [11] and neural networks based techniques [12]. In literature we also find some key techniques in this research area like use map-reduce [13], statistical approaches, text semantics [14] and simple word frequency. The tf-idf weighting technique has been used for keywords extraction before but with different approaches and with different domains of applications, like in support vector machines [15], in website clustering [16] and in automatic key phrases extraction KEA [17] a supervised algorithm where it was used in conjunction with a Bayesian classifier thou it did not use a controlled vocabulary, but instead chose keyphrases from the text itself.

In this study a simpler unsupervised tf-idf based algorithm was proposed in contrasts to Rapid automatic keyword extraction REKA [18], an extended TextRank algorithm [9] which uses three metrics for weighting. The TextRank algorithm is among the popular pioneer in the field in which text is modeled as a social graph with each word represented by a node; the edges can be lexical, semantic or any other relationship required by particular application. For keywords extraction lexical units that co-occurred within a window of N words were used. The pre-processing step in TextRank work involved collapsing all sequences of adjacent keywords into a multi-word keyword. In the proposed approach the opposite is done by pruning chunks of the original text while preserving the word sequence order. The text is split in chunks of n-gram based on the normal punctuations and an a priori chosen set of stop-words in manner described in REKA, However different to REKA approach a simpler final keyphrases ranking scheme that is based on a single weighting metric is used. The top n keywords from the tf-idf scores adapted for a single document are used for back-filtering to retain only the relevant keyword out of the pool of all potential key words.

2.3. TF-IDF for Single Document

The tf-idf is a numerical statistic that is intended to reflect how important a word t is to a document d in a collection or corpus D; mathematically it can be expressed as

$$tfidf = \frac{n_t}{n_d} \; x \; log \frac{D}{D_t}$$

Where

n_t = number of times that term *t* occurs in document *d*

n_d = number terms in document *d*

D = the total number of documents

D_t = the number of documents containing term *t*

The second part of the formula is the inverse document frequency, which measure whether the term is common or rare across all documents.

Although the original design is to work with different documents, tf-idf can be adapted by regarding each sentence as a one-sentence document just as TextRank is the implementation of the general PageRank algorithm to a single document. The single document tf-idf adaption is necessary in ensuring the analysis is not affected by ever changing document streams in a corpus [5]. The tf-idf score in this context will reflect how important the word t is to sentences in that particular single document. In the proposed algorithm the maximum score for the word t is chosen when word t appears in more than one sentence. In that way, the same tf-idf discriminative power for different documents can be harnessed by identifying the top n discriminating terms in the context of a single document.

3. The Algorithm Description

In this section we describe the proposed algorithm. The major strength is in its novel candidate-phrases chunking and the simplicity of the tf-idf weighing scheme adapted to a single document. To begin with, an input document was pre-processed by normalizing all character cases to lower case, then all known English language contractions like "i'm", "ain't" and others were replaced by their normal equivalents. Finally the common English word like 'i', 'me', 'my' and others which carry little or no meaning (stop words) were filtered in additional to domain specific stop words. The pre-processed document was then used to calculate the tf-idf scores according to equation. A different preprocessing was done on the original document in which the stop words and the punctuations are leveraged for phrases splitting using regular expressions. The text chunking is done by taking care of duplicates, empty strings and white spaces while maintaining the order of text chunks sequences from the original document, this is what we call forward word pruning. At that stage then the tf-idf weighting for each text chunk can be done and finally be ranked. The detailed description of the algorithm is given below.

The forward word pruning algorithm

Input: Text document

Output: Ranked keywords list of at most 3-gram

S = *{*StopWords*}*, D = *{*Documents*}*, P = *{*Punctuations*}*

$\forall di \in D$

Begin

(1) Read *di,S,P*

(2) $d_i' \leftarrow di - S$

(3) *punct* ← δ, $\delta \in P$
(4) for *word* $\in di$
  a. if *word* $\in S$
  b. *punct* ← *word*
  c. for *punctuation* $\in P \subset di$
    i. if *punctuation* $\neq \delta$
    ii. *punct* ← *punctuation*
  d. return list d_i'' of *text_chunks*
(5) for *word* $\in d_i'$
  a. *tfidf_score* ← *tfidf* (*word*)
  b. if *word* $\in d_i'$ and *tfidf_score*|*word* = {*s*1, *s*2,… *sn*}
  c. *tfidf_score*|*word* ← *max*{*s*1, *s*2,… *sn*}
(6) for each *text_chunk* in d_i''
  a. if *len*(*text_chunk*) < 3 and if *word* in top n *scores*
  b. split text_chunk into tokens
  c. *text_chunkscore* ←$\sum tfidf\_score\ |_w$,*w* ∈ *text_chunk*
  d. *keyphrase_score* ← *text_chunkscore*
  e. Return the ranked list of the keyphrases

End

Since the tf-idf called in step 5 implicitly employs a loop, as a rule of thumb the proposed algorithm sequence will be dominated by maximum two loops making the algorithm complexity of $O(n^2)$.

4. Experimental Result

The algorithm was implemented in python. The keywords were limited to at most 3-gram basing on the common practice in abstract keywords writing. The filtering result was done on the same abstract sample document from INSPEC database as used by Milhacea for comparison purpose. The results of the initial text chunking and the top-n tf-idf scores are given in Figure 1 and 2, respectively. The proposed tf-idf based weighting scheme for the sample document used is presented in Figure 3; in that figure the candidate key phrases are set against the individual component term. To measure the performance of the proposed scheme it was appropriate to gauge it against the original human keyphrases assigner. A quick Analysis of the original papers[19] abstract revealed that there were keywords and word sequences introduced that never appeared in the given abstract; example the words "homogeneous", "truncated", the sequence "diophantine constraints" and others highlighted by underlining in Figure 5. This signify that manually assigned keyphrases are potentially subjective, this is a drawback since algorithms which conserve the original word sequences order will never reproduce them. So in comparison to the author's manual assignment, all words implicitly added from human expert knowledge domain were ignored and only those which were in the input document were used for matching. Previous studies using this sample abstract used the uncontrolled terms given by the Inspec database as the manually assigned keyword, for comparison sake the same approach was used, assuming that the order of appearance in the Inspec abstract to be the order of importance.

Figure 4 shows our ranked list of the candidate keywords. The comparison of the first eight ranked phrases to the Inspec database keywords is presented in Figure 5. If only first top nine from our ranking are considered which is one third of the unique tokens in the sample our algorithm reproduces five true-positives, matching exactly those given in the Inspec database. This is equivalent to a precision of 63%. Precision is the number of correct results

divided by the number of all returned results, here expressed in percentage. Other used performance measures are recall and F-measure; recall is the number of correct results divided by the number of results that should have been returned while F-measure is given by

$$F - measure = 2\text{x}((precision\ x\ recall)/(precision\ +\ recall))$$

For our result recall is 71% while the F- measure is 67%. The keyword "upper bound" was missed in our top seven keyphrases and also the keyphrases "minimal supporting set" and "minimal set" were generated but did not appear in Inspec databases seven keyphrases. Comparison with REKA which uses three different metric weighting schemes gave the precision of 89% on the same sample input abstract

['compatibility ', ' systems ', ' linear constraints ', ' set ', ' natural numbers', ' criteria ', ' system ', ' linear diophantine equations', ' strict inequations', ' nonstrict inequations ', ' upper bounds ', ' components ', ' minimal set ', ' solutions ', ' algorithms ', ' construction ', ' minimal generating sets ', ' corresponding algorithms ', ' constructing ', ' minimal supporting set ', ' solving ']

Figure 1. The Keywords Candidates Chunked by Stop-words and by Normal Punctuations

[('inequations', 0.29263), ('set', 0.27799), ('systems', 0.27799), ('linear', 0.25597), ('compatibility', 0.25597), ('solutions', 0.25597), ('minimal', 0.25597), ('natural', 0.22992), ('numbers', 0.22992), ('constraints', 0.22992), ('system', 0.18904), ('criteria', 0.16289), ('considered', 0.14631), ('diophantine', 0.14631), ('strict', 0.14631), ('nonstrict', 0.14631), ('equations', 0.14631), ('algorithms', 0.12798), ('upper', 0.11496), ('generating', 0.11496)]

Figure 2. The Single-document Tf-idf Scores for the Pre-processed Token of the Input Text

	'algorithms'	'bounds'	'compatibility'	'components'	'constraints'	'constructing'	'construction'	'corresponding'	'criteria'	'diophantine'	'equations'	'generating'	'inequations'	'linear'	'minimal'	'natural'	'nonstrict'	'numbers'	'set'	'sets'	'solutions'	'solving'	'strict'	'supporting'	'system'	'upper'	'systems'
' algorithms '	0.2																										
' components '				0.1																							
' constructing '						0.2																					
' construction '							0.1																				
' corresponding algorithms '								0.2																			
' criteria '									0.2																		
' linear constraints '					0.2									0.3													
' linear diophantine equations'										0.2	0.2			0.3													
' minimal generating sets '												0.1			0.3					0.1							
' minimal set '															0.3				0.3								
' minimal supporting set '															0.3				0.3					0.2			
' natural numbers'																0.2		0.2									
' nonstrict inequations '													0.3				0.2										
' set '																			0.3								
' solutions '																					0.3						
' solving																						0.2					
' strict inequations'													0.3										0.2				
' system '																									0.2		
' systems '																											0.4
' upper bounds '		0.1																								0.1	
'compatibility '			0.3																								

Figure 3. Tf-Idf based Weighting Scheme for the Sample Document

[(' minimal supporting set ', 0.68027), (' linear diophantine equations', 0.54859), (' minimal set ', 0.53396), (' linear constraints ', 0.48589), (' minimal generating sets ', 0.48589), (' natural numbers', 0.45984), (' strict inequations', 0.43894), (' nonstrict inequations ', 0.43894), (' systems ', 0.3538), (' corresponding algorithms ', 0.3092), (' set ', 0.27799), ('compatibility ', 0.25597), (' solutions ', 0.25597), (' upper bounds ', 0.22992), (' system ', 0.18904), (' criteria ', 0.16289), (' algorithms ', 0.16289), (' constructing ', 0.14631), (' solving ', 0.14631), (' components ', 0.11496), (' construction ', 0.11496)]

Figure 4. The Ranked Key Phrases Output

Title : Compatibility of systems of linear constraints over the set of natural numbers

Abstract:

Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types

Author's key words :

system of linear diophantine equations, homogeneous linear Diophantine equations, nonhomogeneous linear diophantine equations, system of linear diophantine constraints, homogeneous linear diophantine constraints, nonhomogeneous linear diophantine constraints, linear diophantine inequations, homogeneous linear diophantine inequations, criteria of compatibility, truncated set of solutions

Inspect database keywords (Uncontrolled terms):

linear constraints - set of natural numbers - linear Diophantine equations - strict inequations - nonstrict inequations - upper bounds - minimal generating sets

Figure 5. The Original Sample Abstract with the Author's Assigned Keywords and the Ones Assigned in the Inspec Database

Table 1. Comparison of the Generated Key Phrases from the Proposed Algorithm with the Sample Abstracts Author Manual Key Phrases and Inspec Database Manual Key Phrases

Manual - Author	*Manual – Inspec database*	*Rank*	*Ours-TF-IDF based*	*Rank*
system of linear diophantine constraints , homogeneous linear diophantine constraints , nonhomogeneous linear diophantine constraints	*linear constraints*	*1*	*' linear constraints '*	*4*
	set of natural numbers	*2*	*' natural numbers '*	*6*
system of linear diophantine equations , homogeneous linear Diophantine equations , nonhomogeneous linear diophantine equations	*linear Diophantine equations*	*3*	*linear Diophantine equations*	*2*

linear diophantine inequations , homogeneous linear diophantine inequations	*strict inequations*	*4*	*' strict inequations'*	*7*
	nonstrict inequations	*5*	*' nonstrict inequations '*	*8*
	upper bounds	*6*		
	minimal generating sets	*7*	*minimal generating sets*	*5*
			minimal supporting set	*1*
			minimal set	*3*
criteria of compatibility				
truncated set of solutions				

5. Extension to YouTube Videos

We researched the usefulness of our algorithm by applying it to automatic generation of the YouTube how-to videos [2] which we are convinced are potential learning sources for many people who seek education informally. A python based YouTube metadata extraction API was used to get four parts of interest from the metadata namely the title, the author and the video description and the author assigned keywords. A typical YouTube video metadata like in Figure 6 has other parameters like number of views, likes, length, date of publication *etc.*, some of which (like keywords) are hidden from the user

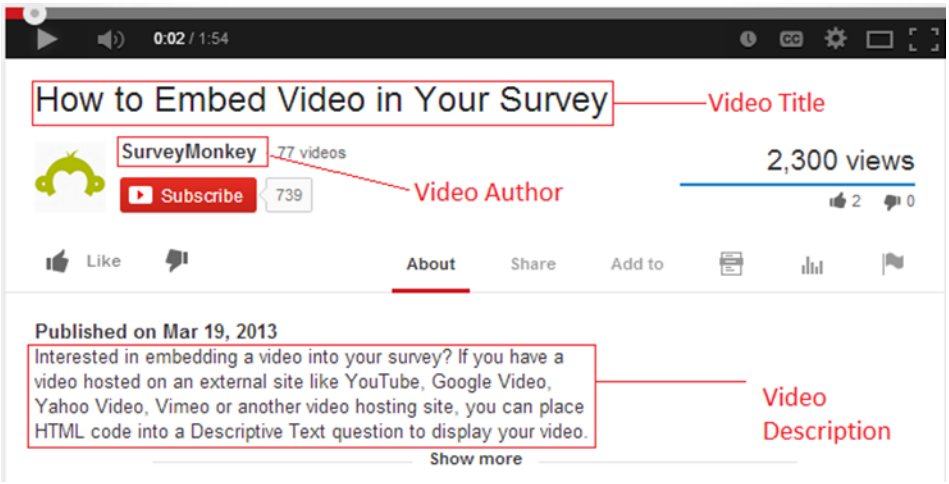

Figure 6. Part of YouTube's Video Metadata

In this metadata retrieval sample, the author the title and the video were concatenated to form a single text document which was tested. The sample result can be seen in Figure 7 below. As noted earlier, author-assigned keywords will show some biasness to his or her knowledge domain; in this case also some words were noted that were not part of the title or description for example the phrase 'Nyan Cat'. The other keyword, 'survey monkey',' embed video' and 'html code' were captured well disregarding the order of their appearances. The proposed keywords on the YouTube sample video are objective in nature; they might prove useful in objectively annotating YouTube videos especially the instructional and educational videos like the How to video.

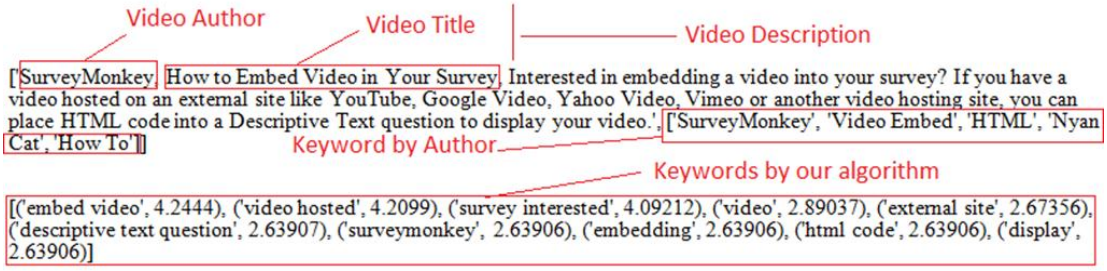

Figure7. Extracted Metadata and our Generated Keyword based on Author, Title and Description

6. Discussion

The study objective was to demonstrate the simpler tf-idf based keywords extraction approach; to show the extent of subjectivity of the annotations in the UGCs and show how the proposed algorithm might help in solving such issues. The algorithm description was given and the comparative performance on the sample text used in previous studies is encouraging. The extension of the algorithm application to YouTube's metadata shows how useful it can be in dealing with subjectivity in tagging. The issue of subjectivity UGC tagging was explore with evidence presented using the YouTube video author keywords assignment in section 5 and the popular sample abstract used as a standard in famous keyword algorithms like TextRank and REKA in section 4. Our investigation of the original manuscript revealed that in the first case, the keywords assigned by Inspec experts were different from the original author's assignment. Comparison to the original paper's [16] abstract revealed that the author chose as keywords a sequence of words that never appears in the body of the given abstract; for example the words '*homogeneous', 'truncated', 'diophantine constraints'* and others highlighted by underlining in Figure 5 were subjectively introduced by the content creator. In the case of YouTube sample video, the author assigned keywords 'Nyan Cat' and 'How to', the first one is not found anywhere in the metadata text extracted while the second one constitutes two common English stop-words that will always be filtered in many keywords extraction algorithms and thus can never be reproduced. User assigned keyphrases especially in UGC domain are potentially subjective relative to their domain of expertise, locality and culture. Objective keyword extraction should only be gauged basing on the given text content or metadata otherwise modeling becomes complex. In both tested the proposed approach objectively captures the important keyword and thus it can be used in indexing applications or help experts and non-expert users who might need automation in tagging their created digital content.

7. Conclusion

The importance of discovery of multimedia content in the context of web 2.0 cannot be over-emphasized. Text mining and natural language approaches are well established so it is always preferred to cast discovery problem into those domains for effective retrieval. In this study, a simpler non graph based algorithm that exploits the power of tf-idf weighting scheme for single document to extract keyword was proposed. Also the challenges of modeling human tagging were demonstrated by presenting the original authors' assigned keyphrases. The performance of the proposed approach was reasonable though little less that of the three metric based weighting scheme REKA. The applicability of the algorithm is seen in sample experimental result on YouTube videos. Our major contribution is twofold; the simplicity of

our proposed automatic keyword extraction approach and the evidence based discourse of the UGC creator's subjectivity in tagging. Our keywords extraction method is an approach towards solving that bias.

References

[1]. P. Bellini, D. Cenni and P. Nesi, "International Journal of Multimedia Information Retrieval", (**2014**). pp. 1-13.
[2]. A. R. Mtafya and D. Huang, "International Journal of Digital Content Technology and its Applications", vol. **7**, no. 12, (**2013**). pp. 56 - 63.
[3]. B. Croft, D. Metzler and T. Strohman, "Search Engines: Information Retrieval in Practice": Addison-Wesley Publishing Company, (**2009**).
[4]. M. Winget, "User-Defined Classification on the Online Photo Sharing Site Flickr...Or, How I Learned to Stop Worrying and Love the Million Typing Monkeys". in 17th Annual ASIST SIG/CR Classification Research Workshop, (**2006**), Austin, TX, United States.
[5]. S. Rose, D. Engel, N. Cramer and W. Cowley, "Automatic Keyword Extraction from Individual Documents", in Text Mining John Wiley & Sons, Ltd, (**2010**), pp. 1-20.
[6]. Z. Zede, L. Miao, C. Lei, Y. Zhenxin and C. Sheng, "Combination of Unsupervised Keyphrase Extraction Algorithms". in IALP 2013, (**2013**), Piscataway, NJ, USA.
[7]. S. Lahiri, S. R. Choudhury and C. Caragea, "arXiv preprint arXiv:1401.6571", (**2014**).
[8]. Z. Liu, X. Chen, Y. Zheng and M. Sun. "Automatic Keyphrase Extraction by Bridging Vocabulary Gap", in 15th Conference on Computational Natural Language Learning, (CoNLL 2011), (**2011**), Portland, OR, United States.
[9]. R. Mihalcea and P. Tarau, "Textrank: Bringing Order into Texts", in Proceedings of EMNLP-04 and the 2004 Conference on Empirical Methods in Natural Language Processing, (**2004**), Barcelona, Spain.
[10]. L. Zhang and Y. Tang, "Keywords Extraction Method Based on Deleting Actor Index", in Computer Science & Education (ICCSE), 2013 8th International Conference on, (**2013**), Colombo, Sri Lanka.
[11]. K. Lei, H. Tang and Y. Zeng, "Keywords Extraction Via Multi-Relational Network Construction", in 3rd International Conference on Computational Science, Engineering and Information Technology (CCSEIT 2013), (**2013**), Konya, Turkey.
[12]. B. Zyglarski and P. Baa, "Keywords Extraction: Selecting Keywords in Natural Language Texts with Markov Chains and Neural Networks", in International Conference on Knowledge Management and Information Sharing, (KMIS 2010), (**2010**), Valencia, Spain.
[13]. A. Nugumanova, A. Novosselov, Y. Baiburin and A. Karimov, "Automatic Keywords Extraction from the Domain Texts: Implementation of the Algorithm Based on the Mapreduce Model", in 2013 International Conference on Current Trends in Information Technology, (CTIT 2013), (**2013**), Dubai, United Arab Emirates.
[14]. M. Han, X. Zhang and X. Wang, "Journal of Information and Computational Science", vol. 6, no. 3, (**2009**), pp. 1495-1503.
[15]. K. Zhang, H. Xu, J. Tang and J. Li, "Keyword Extraction Using Support Vector Machine", in Advances in Web-Age Information Management, J. Yu, M. Kitsuregawa, and H. Leong, Editors Springer Berlin Heidelberg, (**2006**), pp. 85-96.
[16]. P. Tonella, F. Ricca, E. Pianta and C. Girardi, "Using Keyword Extraction for Web Site Clustering", in Web Site Evolution, 2003, Theme: Architecture Proceedings Fifth IEEE International Workshop on, (**2003**), Amsterdam, The Netherlands.
[17]. I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin and C. G. Nevill-Manning, "Kea: Practical Automatic Keyphrase Extraction", in Proceedings of the fourth ACM conference on Digital libraries, (**1999**), Berkeley, California, USA.
[18]. S. J. U. S. Rose, W. E. U. S. Cowley, V. L. U. S. Crow and N. O. U. S. Cramer, "Rapid Automatic Keyword Extraction for Information Retrieval and Analysis" U.S. Patent 8131735, (**2012**).
[19]. S. L. Kryvyi, "Cybernetics and Systems Analysis", vol. 38, no. 1, (**2002**), pp. 17-29.

