*Scientia Africana, Vol. 13 (No.2), December 2014. Pp53-68*
 *ISSN 1118 - 1931*

# AN ENHANCED MODEL FOR AUTOMATICALLY EXTRACTING TOPIC PHRASE FROM WEB DOCUMENT SNIPPET FOR CLUSTER LABELS

**C. I. Ejiofor [1], E. O. Nwachukwu [2] and E. E. Williams [3]**

[1,2]*Department of Computer Science,*
*University of Port-Harcourt, River State, Nigeria,*

[3]*Department of Computer Science*
*University of Calabar, Cross River State, Nigeria*

[1]ejioforifeanyi@yahoo.com [2]enoch.nwachukwu@uniport.edu.ng [3]edemwilliam@yahoo.com

*Received: 20-02-14*
*Accepted: 19-05-14*

## ABSTRACT

*Keyphrase are subset of more than one word or phrases from a document that can describe the meaning of the document. Manual assignment of high quality document into similar topic by keyphrase is expensive, time-consuming and error prone. Therefore, various unsupervised ranking methods based on importance scores are proposed for keyphrase extraction. There are two approaches for keyphrase-based categorization: manual and automatic. In the manual approach, a human expert performs the classification task, and in the second case, supervised classifiers are used to automatically classify resources. In a supervised classification, manual interaction is required to create some training data before the automatic classification task can takes place. In our new approach, we propose automatic classification of documents through semantic keyphrase and a new model for generating keyphrase for web document topic label. Thus we reduce the human participation by combining the knowledge of a given classification and the knowledge extracted from the data. The main focus of this paper is the automatic classification of documents into machine-generated phrase-based cluster labels for classifications. The key benefit foreseen from this automatic document classification is not only related to search engines, but also to many other fields like, document organization, text filtering and semantic index managing.*

**Key words:** Keyphrase extraction, machine learning, search engine snippet, document classification, topic tracking

## INTRODUCTION

Automatic keyword extraction (AKE) is the task to identify a small set of words, key phrases, topics, keywords, or key segments from a document that can describe the meaning of the document [1]. Since keyword is the smallest unit which can express the meaning of document, many text mining applications can take advantage of it, for automatic indexing, automatic summarization, automatic classification, automatic clustering, automatic filtering, topic detection and tracking, information visualization, etc. Therefore, keywords extraction can be considered as the core technology of all automatic processing for web documents retrieval and classification. The present information explosion increases the importance of this area. It is difficult to find out relevant information from a huge

information mass like the Internet [28], but this research makes it possible and easy.

However, over the last 30 years, extensive work has been done in the area of Machine Learning. Whether the approach is inspired by biological plausibility (Artificial Neural Networks) or by psychological plausibility (Artificial Intelligence), the goal is to design systems capable of learning and reasoning about certain tasks such as phrase or text extraction, classification, clustering e.t.c. Most existing systems focus on some observed aspect of human learning, and attempt to match, and possibly exceed, human abilities. For example, several inductive models exists that exhibit similar (and sometimes better) predictive accuracy than their human counterparts on various problems. It is the system's ability to capture and exhibit certain important characteristics (often inspired by human learning), that allows it to be useful as an artificial learning system or agent.

As machine learning aims to address larger, more complex tasks, the problem of focusing on the most relevant information in a potentially overwhelming quantity of data has become increasingly important. The Internet and World Wide Web have put a huge volume of low quality information at the easy access of learning system. For instance, data mining of corporate or scientific records often involves dealing with both many features and many examples for the use in representing data and the problem of selecting the most relevant examples to drive the learning process. At a conceptual level, one can divide the task of concept learning into two subtasks: deciding which features to use in describing the concept and deciding how to combine those features, people are increasingly using the Web to learn an unfamiliar topic because of the Web's convenience and its abundance of information and knowledge.

Traditional approaches to information retrieval, which are popular in current search engines, use direct keyword matching between documents and query representations in order to select relevant documents. The most critical point goes as follows: if a document is described by a keyword different from those given in a query, then the document cannot be selected although it may be highly related. This situation often occurs in real cases as documents are written and sought by different persons [28]. In many cases, this traditional method of learning may not even be applicable because in our fast changing world, many topics and technologies emerge constantly and rapidly. There is often not enough time for someone to compile all the existing knowledge and to make contributions in the current state of the art of a research topic.

In recent work, there is common agreement that more adequate relevance estimation should be based on inference rather than direct keyword matching [9], [21], [38], [40]. That is, the relevance relationship between a document and a query should be inferred using available knowledge. This inference, however, cannot be performed with complete certainty as in classical logic due to the uncertainty inherent in the concept of relevance: one often cannot determine with complete certainty if a document is relevant or not. In Information Retrieval (IR), uncertainty is always associated to the inference process [28]. In order to deal with this uncertainty, probability theory has been a commonly used tool in IR [29], [23], [32], [41].

*Scientia Africana, Vol. 13 (No.2), December 2014. Pp53-68*
*© College of Natural and Applied Sciences, University of Port Harcourt, Printed in Nigeria* *ISSN 1118 - 1931*

Probabilistic models usually attempt to determine the relationship between a document and a query through a set of terms that are considered as features. Within the strict probabilistic framework, inferential approaches are often confined to using only statistical relations among terms. The main method adopted by probability theory to determine the relevance degree among terms is by considering term co-occurrences in the document collection [36]. In this case, two terms which often co-occur are considered strongly related. The problem stands out in this method because relations obtained from statistics may be very different from the genuine relations: truly connected terms may be overlooked [33] whereas truly independent terms may be put in relation [14].

A new method using human-defined knowledge like a thesauri to establish the relationship among terms is now getting popular in IR. With the recent development of large thesauri (for example, Wordnet [25]), these relations have quite a good coverage of application areas. A manual thesaurus is then a valuable source of knowledge for IR. However, due to the lack of strict quantitative values of such relations in thesauri, the quantitative values have to be determined by user relevance feedback or expert training.

**An Overview of Phrase Extraction Techniques**

The field of text mining seeks to extract useful information from unstructured textual data through the identification and exploration of interesting patterns. The techniques employed usually do not involve deep linguistic analysis or parsing, but rely on simple "bag-of-words" text representations based on vector space. Natural Language Processing (NLP), Data Mining, and Machine Learning techniques work together to automatically classify and discover patterns from the electronic documents. The main goal of text mining is to enable users to extract information from textual resources and deals with the operations like, retrieval, classification (supervised, unsupervised and semi supervised) and summarization.

Many methods have been proposed for keyphrase extraction, most of them are based on machine learning techniques which is done systematically and with either minimal or no human intervention, depending on the model. In this approach, phrases are extracted from documents and are labeled as keyphrases or non-keyphrases. The documents and labeled phrases are then used as training data for creating a keyphrase classifier. The goal of automatic extraction is to apply the power and speed of computation to the problems of access and discoverability, adding value to information organization and retrieval without the significant costs and drawbacks associated with human indexers.

Ever since the inception of the Web, searching and extracting useful information from it has been an active research area. So far, many information extraction techniques have been proposed and some of them are also widely used in practice. These techniques include keyword-based search, phrase-base extraction, wrapper information extraction, Web queries, user preferences, and resource discovery. Keyword-based search using search engines [4] is clearly insufficient for our task as discussed in the Introduction section. Wrapper-based approaches [1], [7], [11], [12] are not suitable either because Wrappers basically

help the user extract specific pieces of information from targeted Web pages. Hence, they are not designed for finding salient concepts and definitions of user specified topics, which can be of any type. Web query languages [24] allow the user to query the Web using extended database query languages. They are also not suitable for our problem. In the user preference approach (used commonly in push type of systems [39], information is presented to the user according to his/her preference specifications. This is clearly inappropriate for our problem. Web resource discovery aims to find Web pages relevant to users requests or interests [10], [18], [19], [27]. This approaches, uses techniques such as link analysis, link topologies, text classification methods to find relevant pages. The pages can also be grouped into authoritative pages, and hubs. However, relevant pages, which are often judged by keywords, are not sufficient for our purpose because we need to further process the contents of the Web pages to discover those salient concepts of the topic and descriptive pages by extracting keyphrase from web document that represent the salient concept of the topic.

**Existing Techniques to Keyphrase Extraction**

The manual extraction of keyphrase is slow, expensive and bristling with mistakes. Therefore, most algorithms and systems to help people perform automatic keyphrase extraction have been proposed. Existing methods can be divided into four categories: machine learning, simple statistics, linguistics, and mixed approaches [30], [44].

**Machine Learning Techniques**

Keyphrase Extraction can be seen as supervised learning, Machine Learning approach employs the extracted keywords from training documents to learn a model and applies the model to find keyphrases from new documents[16], [20], [37].

**Simple Statistics Techniques**

These methods are simple, have limited requirements and do not need the training data. They tend to focus on non-linguistic features of the text such as term frequency, inverse document frequency, and position of a keyword. The statistics information of the words can be used to identify the keywords in the document. Cohen uses N-Gram statistical information to automatically index document [8]. Other statistics methods include word frequency, TF*IDF, word co-occurrences [22], etc. The benefits of purely statistical methods are their ease of use and the fact that they do generally produce good results.

**Linguistics Techniques**

These approaches use the linguistic features of the words mainly sentences and documents. The linguistic approach includes the lexical analysis, syntactic analysis discourse analysis and so on [26], [30], [43]. In fact, some of the linguistic methods are mixed methods, combining some linguistic methods with common statistical measures such as term frequency and inverse document frequency [15].

**Hybrid Techniques**

Other approaches about keyword extraction mainly combine the methods mentioned above or use some heuristic knowledge in the task of keyword extraction, such as the position, length, layout feature of the words, html tags around of the words, etc [17]. The overview of the related works reveals that the automatic keyword extraction is faster

*Scientia Africana, Vol. 13 (No.2), December 2014. Pp53-68*
*© College of Natural and Applied Sciences, University of Port Harcourt, Printed in Nigeria* *ISSN 1118 - 1931*

and less expensive than human intervention. However, currently existing solutions for automatic keyword extraction require either training examples or domain specific knowledge. Our approach, on the contrary, does not have this additional information. We apply the statistical measures to automatically extract keyword as they are domain independent and have limited requirements. Moreover, in this research we tried analyse how the database context can be exploited in order to automatically extract representative keywords from a document.

**Keyword Extraction Systems**

**Kea:** Kea is a keyword extraction system which implemented their methodology with an algorithm for automatically extracting keyphrases from the text. [42] Kea identifies candidate keyphrases using pre-processing [13], calculates feature value for each candidate, and uses a machine learning algorithm to predict which candidates are good keyphrases. The Naïve Bayes machine learning scheme first builds a predication model using the training documents with known keyphrases, and then uses the model to find keyphrases in new documents. Two features are calculated for each candidate phrase and used in training and extraction. They are: TF × IDF, a measure of phrase's frequency in a document compared to its rarity in general use and first occurrence, which is the distance into the document of the phrase's first appearance. Kea´s effectiveness has been assessed by counting the keyphrases that were also chosen by the document's author, when a fixed number of keyphrases are extracted.

**Barker Approach:** Barker [2] describes a system for choosing noun phrases from a document as keyphrases. A noun phrase is chosen based on its length, its frequency and the frequency of its head noun. Noun Phrases are extracted from a text using a base noun phrase skimmer and an off-the-shelf online dictionary. Barker approaches involves human judgment for performing this experiments.

**KPSpotter**: KPSpotter is an algorithm implementing the methodology proposed by [34]. After classical pre-processing has been applied, the algorithm employs a technique that combines information gain and a data mining measure technique introduced in [35]. In this sense KPSpotter presents some resemblances with extractor. Both algorithms, in fact, use a learner belonging to the same family, i.e., the decision tree [35].

**Our Approach**

In this paper, we present a web-based phrase extraction technique for web search document clustering result, with rich features for giving users the flexibility of retrieving closely related documents clustered in user friendly way based on his search query. Searching the web and developing search tools has not been easy; as there are complicated tools and algorithms involved in designing an interface that will give users the best possible and effective results. The description of our web-based phrase extraction technique for web search document clustering result interface is described in line with its mathematical model. Our topic phrase extraction technique is wrapped around the Google search engine result using its API as its primary source of retrieving search results from the Internet.

The Google search engine is currently ranked among the best search engines in the world (Alsulami et al., 2012), hence its use in this research. Consequent upon this ranking, we assume that Google will provide us with highly relevant-to-topic ranked documents snippet. So we are concerned with how to extract topic phrase that can be used to form cluster labels for search results in order to make it easier for the user to wade through the results in a most effective manner to get what he is searching for. So we are not going to be primarily involved in the mechanism of pages retrieval and ranking, but rather with that of phrase topic extraction from Google result snippet as cluster label.

**Extracting Topic Phrase Table from Google Result Snippet**

The idea of clustering is to categorize the results set from the search engine into topics which the user can easily wade through and grouping the pages according to topics that are closely related to the users search intention.

One method for phrase-based topic extraction from search engine result is to use proximity measures to specify the acceptable distance between words. Two or more words are considered to be a phrase if they are separated by no more than N words in the web document snippet description text, for example if N=0, then the words must be adjacent if N=1 then they can have only one word between them. The topic phrase labels are formed using snippet of a search engine result. In this paper we made use of phrases extracted from snippet of the first top 100 pages of a search engine ranked result, TP100, to form the topic labels.

Researchers have determined experimentally that the first 100 pages of search engine result always contain the most relevant-to-search-query documents.

Figure 1 illustrates the process of extracting keyword topic from web search result for document clustering. Starting with a user querying a search engine with a phrase “jaguar”, a list of relevant documents is found by matching strings in the search phrase with documents. While most search engines stop after the second step (the actual retrieval of document) for users to sift through thereby causing information overload, topic clustering proceeds by analyzing the documents in the result list in order to group document to similar topics. The user who queries for “jaguar” can then choose among the clusters and access the corresponding documents as shown in Figure 1.

*Scientia Africana, Vol. 13 (No.2), December 2014. Pp53-68*
*© College of Natural and Applied Sciences, University of Port Harcourt, Printed in Nigeria* *ISSN 1118 - 1931*

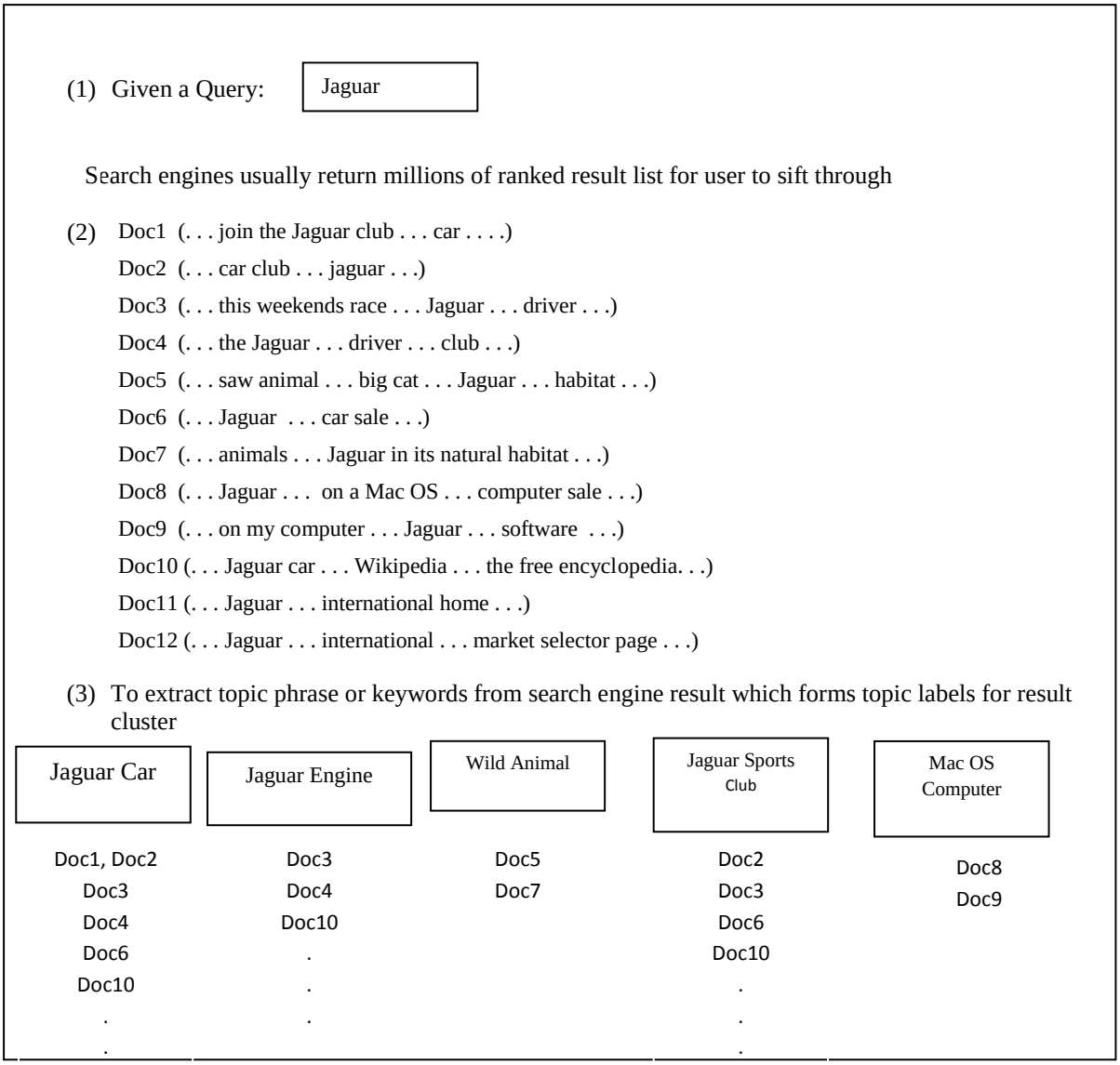


Fig. 1 An illustration of topic phrase extraction from web search result for document cluster label

**Transforming Web Documents Snippet into Vector Representation**

Transforming of text documents to real vector is an essential step for text mining tasks such as classification, clustering and information retrieval. The extracted vectors, serves as inputs for data mining models, and suppose that there are primitive concepts (basis concepts or topics) in a document space which can be used to form the concepts or topics used in the field of information retrieval. In order to find out primitive concepts or topics, we assume that web documents snippet contain features that characterize the primitive concepts or topics.

We have two steps to make primitive concepts or topics

i. To extract topic phrases as feature from a web document snippet by selecting significant

phrases and partition significant phrases into feature vectors

ii. To cluster the feature into primitive concepts or topics

**Document Preprocessing**

There are several steps to preprocess documents in order to convert them into suitable terms that describe better content of a web document to a structured form as vector representation. At this stage, we typically use a combination of three common text-preprocessing methods:

**Stemming:** Stemming algorithms are used to transform the words in texts into their grammatical root form, and are mainly used to improve the Information Retrieval System's efficiency. To stem a word is to reduce it to a more general form, possibly its root. For example, stemming the term interesting may produce the term interest. Though the stem of a word might not be its root, we want all words that have the same stem to have the same root. The effect of stemming on searches of English document collections has been tested extensively. Several algorithms exist with different techniques. The most widely used is the Porter Stemming algorithm.

**Elimination of Stop Words**

It is necessary to remove unwanted words after stemming and there are about 400 to 500 types of stop words that can be found in a sentence or title of a document, such as "of", "and", "the," etc., that provide no useful information about the document's topic. Stop-word removal is the process of removing these words. We use a standard list of 571 stop words and remove them from the documents.

**Tokenization:** Tokenize process is used to determine sentence boundaries, and to separate the text into a stream of individual tokens (words) by removing extraneous punctuation. It separates the text into words by using spaces, line breaks, and other word terminators in the language. Document texts must be tokenized correctly in order for the noun phrase extractor to parse the text efficiently. It also works as a basis for phrase boundary detection.

**Extracting Topic Feature Vector from a Web Document Snippet**

In this study, we defined the term "topic" as a stream of terms or phrases which represent the content of the web document. A topic is different from a title, while the later is a sequence of terms that rather represent the name of a study with a document and does not necessarily represent the concept of the study in the document. Most of the documents are embossed by their snippets; however, the title is not necessarily stands for the content of the web documents and it is not possible to judge about the content of documents by only their titles. The automatic web document topic identification is not an easy task as a web document may contain multiple topics.

In order to extract document topic feature (in a form of vector) from a snippet, we used the pseudocde as described in Figure 2 with the following hybrid models of tf-idf, title term frequency method and Query-Bias method to determine candidate topic feature for cluster label.

The pseudocode is given in Figure 2

```
for each document d(i) in the Top 100 documents
    scan snippet of d(i) for keywords ;
```

*Scientia Africana, Vol. 13 (No.2), December 2014. Pp53-68*
*© College of Natural and Applied Sciences, University of Port Harcourt, Printed in Nigeria* *ISSN 1118 - 1931*

```
for each keyword K(j) in document d(i) snippet
        if keyword exists in keywords table then
            increment F k(i,j) ;// frequency counter of k(j) in d_i
            increment GF k(j);// global frequency of the keyword in the search engine result set
        else // if keyword is not already in the keywords table
            add keyword k(j) to the keywords table;
            set counter F K(j)=1;
            increment document count D k(j); //count the no. of documents in which keyword occurs
        end if
    next keyword;
next document;
```

**Fig. 2: Pseudocode**

**Proposed Method**

In this step, the stems of the words are used as the dimensions or the features of the vector space model. In this model, each document d is considered as a vector in a word space. It is shown by the word frequency vector. This vector is shown in Table 1. Implanting vector space model for a set of documents retrieved as a result of web search consist of transforming the string of words that make up a web document snippet into equivalent numeric vector. The vectors will have the same size, turning our result into m x n matrix. Each line represents one web page and each column represents one word. The N dimension represents the total number of words that will be processed from the first k result of a search engine ranked list. The M dimension represents the remaining number of web pages after the duplicate eliminations, of the final search result list.

The first step in implementing the vector space model consists of the construction of the index term vector. The index term vector is constructed from all the individual unique words of the web document snippet and title of the retrieved documents. Each of the retrieved result will be divided into component words and added to the index vector.

The second step consist of document vectorisation: each document will be transformed into its numeric vector representation. In the vector space model, each document will be represented by a numeric vector of length n, where n is the dimension of the index term vector. The value of each word or phrases in the document corpus will be calculated with hybrid formula: Chen's algorithm (1998), title term frequency method and Query-Bias method.

**Table 1: Vector space representation of a search engine snippet for a query**

| $Document_m$ ($D_m$)/$Term_n$ ($T_n$) | $D_1$ | $D_2$ | $D_3$ | - - - | $D_m$ |
|---|---|---|---|---|---|
| $T_1$ | $TF_{11}$ | $TF_{12}$ | $TF_{13}$ | - - - | $TF_{1m}$ |
| $T_2$ | $TF_{21}$ | $TF_{22}$ | $TF_{23}$ | - - - | $TF_{2m}$ |
| $T_3$ | $TF_{31}$ | $TF_{32}$ | $TF_{33}$ | - - - | $TF_{3m}$ |
| - - - | - - - | - - - | - - - | - - - | - - - |
| $T_n$ | $TF_{n1}$ | $TF_{n2}$ | $TF_{n3}$ | - - - | $TF_{nm}$ |

Given a finite set D of elements called web documents

$D = \{D_1 - - - - D_j - - - - D_m\}$

And a finite set T of elements called index terms

$T = \{t_1 - - - - t_i - - - - t_n\}$

Any document $D_j$ is assigned a vector $V_j$ of finite real numbers called weights of length m as follows

$V_j = (w_{ij})_i = 1, - - - -, n = (w_{1j}, - - - -, w_{ij}, - - - - w_{nj})$

Where $0 \leq w_{ij} < 1$(i.e $w_{ij}$ is normalized, eg division by the largest)

The weight $w_{ij}$ is interpreted as an extent to which the term ti characterizes the web document $D_j$ and is computed using the term frequency-inverse document frequency given as

$$W_{ij} = tf_{ij} \times idf_i \quad - - - - - - - - - (1)$$

$$Idf_i = \log_2 n - \log_2 df_j + 1 \quad - - - - - (2)$$

$$W_{ij} = tf_{ij} \times \log_n \frac{n}{df_j} \quad - - - - - (3)$$

Where

$W_{ij}$ = represents the weight of index term $t_i$ from document $D_j$

$Tf_i$ = represents the frequency of term $t_i$

$df_i$ = represents the number of documents in which the term $t_i$ appears

n = total number of documents retrieved by the search engine for a query term $Q_i$

A fragment of the keyphrase table is shown in Table 2.

**Table 2: Computing the weights from the $tf_{ij}$**

| $Document_m$ ($D_m$) | keyphrase | Frequency in web snippet | Document Frequency (df) | Weight ($W_{ij}$) ($tf_{ij} \times idf_i$) |
|---|---|---|---|---|
| $D_1$ | $T_1$ | $n_1$ | $df_{13}$ | $W_{1j}$ |
| $D_2$ | $T_2$ | $n_2$ | $df_{23}$ | $W_{2j}$ |
| $D_3$ | $T_3$ | $n_3$ | $df_{33}$ | $W_{3j}$ |
| - - - | - - - | - - - | - - - | - - - |
| $D_n$ | $T_n$ | $n_n$ | $df_{n3}$ | $W_{ij}$ |

*Scientia Africana, Vol. 13 (No.2), December 2014. Pp53-68*
*© College of Natural and Applied Sciences, University of Port Harcourt, Printed in Nigeria* *ISSN 1118 - 1931*

**Title Terms Frequency Method**

It is generally known that the titles of an article tend to reveals the major subject of that document. This hypothesis was examined in a sample study of TREC documents where the title of each article was found to convey the general idea of its contents. In order to exploit this feature in document collection, terms that occurred in the title section of the documents were assigned a positive weight (title score). Thus, a factor in the document snippet score is the presence of title words within the snippet description. In order to utilise this attribute in the cluster label extraction for topic tracking process, each constituent term in the title section is looked up in the body of the snippet. For each document snippet a title score is computed as follows,

TSS=TTS/TTT

where
TSS = the title score for a web document snippet
TTS = the total number of title terms found in a web document snippet
TTT = the total number of terms in a web document title

TTT is used as a normalization factor to ensure that this method does not have an excessive sentence score factor contribution relative to the overall sentence score.

**Query-Bias Method**

The addition of a snippet score factor bias to score snippets containing query terms more highly may reduce the query drift caused by the use of bad feedback terms. Thus, whether a relevant or non-relevant document is used, the feedback terms are taken from the most relevant title identified in the document snippet, in relation to the submitted query.

In order to generate a query biased label in this work, each constituent snippet of a document being processed is scored based on the number of query terms it contains. The following situation gives an example of this method.

• For a query "falkland petroleum exploration" and
• A snippet "The british minister has decided to continue the ongoing **petroleum exploration** talks in the **falkland** area"

The query score QSS is computed as follows

$QSS = tq^2/nq$

where
tq = the number of query terms present in a web document snippet
nq = the number of terms in a query

Therefore the query score QSS for the above snippet is 3. This score is assigned based on the belief that the number of query terms contained in a snippet, the more likely it is that this snippet conveys a large amount of information related to the query. This was the same method used in [14].

**Combining the Scores**

The previous sections outlined the components used in scoring keyphrases to generate a meaningful label used for cluster in this work. The final score for each topic label is calculated by summing the individual score factors obtained for each method used. Thus the final score for each phrase topic label is

$CLSS = TSS + QSS + W_{ij}$

where
CLSS = Cluster Label Significance Score

**Table 3: Combing Weight of the terms in snippet ($tf_{ij}$) with the Scores**

| $Document_m$ ($D_m$) | Keyphrase ($Term_n$) | Frequency in web snippet | Document Frequency (df) | Weight ($W_{ij}$) ($tf_{ij}$ x $idf_i$) | TSS | QSS | CLSS |
|---|---|---|---|---|---|---|---|
| $D_1$ | $T_1$ | $n_1$ | $df_{13}$ | $W_{1j}$ | $TSS_1$ | $QSS_1$ | $CLSS_1$ |
| $D_2$ | $T_2$ | $n_2$ | $df_{23}$ | $W_{2j}$ | $TSS_2$ | $QSS_2$ | $CLSS_2$ |
| $D_3$ | $T_3$ | $n_3$ | $df_{33}$ | $W_{3j}$ | $TSS_3$ | $QSS_3$ | $CLSS_3$ |
| - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - |
| $D_m$ | $T_n$ | $n_n$ | $df_{n3}$ | $W_{ij}$ | $TSS_m$ | $QSS_m$ | $CLSS_m$ |

The phrase extraction system was implemented such that each method could be invoked independently. Thus it was possible to experiment with various combinations of the methods described above to determine the best keyphrase extraction method(s) for term selection in PRF.

In order to generate an appropriate keyphrase label for clusters, it is essential to use web document snippet as they contain the summary of the entire document. The inspection of our example phrase extraction for cluster label showed them to be reasonable representations of the topic for the original documents. However, in our case an objective measure of cluster label quality is their overall effect on retrieval performance to categories document based on their similarity with the concept for the cluster label.

**Semantic Similarity of Document Concept Based on Snippet Approach**

Our approach uses the web as a corpus and a web search engine such as Google as backend engine. Web search engines index billion of pages on the web and provide an estimate of page counts as a result for a searching term. Most search engines provide an interface to search for a term or more using Boolean operators on document content. In our method, our idea is to find an attribute that is short enough for the co-occurrence to be considered and good enough to describe document's topic and idea. In order to measure semantic similarity between two given terms t1, t2, the proposed approach will search for the terms t1 and t2 in the snippet of the document instead of the content of the document as in the following.

Each document is a vector of terms in snippet as

$D_i = (t_{i1}, t_{i2}, t_{i3}, t_{i4}, ... t_{in})$

$D_j = (t_{j1}, t_{j2}, d_{j3}, t_{j4}, ..., t_{jn})$

To compute the similarity of the topic terms, we

1. Search in search engine snippet for term t1 and Let count (t1), be the number of search engine snippet containing term t1.
2. Search in search engine snippet for term t2 and Let count (t2), be the number of search engine snippet containing term t2.
3. Search in snippet for both terms t1 and t2 and Let count (t1, t2), be the number of snippets containing both terms t1 and t2.
4. Compute scores using count (t1), count (t2) and count (t1,t2).

*Scientia Africana, Vol. 13 (No.2), December 2014. Pp53-68*
*© College of Natural and Applied Sciences, University of Port Harcourt, Printed in Nigeria* *ISSN 1118 - 1931*

The resulting score is a measure of similarity. In order to compute count (t1), count (t2) and count (t1, t2), given two terms t1 , t2 and a similarity function Measure(t1,t2)>0. The general transformation formula of measure(t1,t2) function to snippet similarity is defined as

$$\text{Similarity}\quad (S_i, S_j) = \frac{\sum_{k=1}^{n} t_{ik} * t_{jk}}{\sqrt{\sum_{k=1}^{n} t_{ik}^{\,2} * \sum_{k=1}^{n} t_{jk}^{\,2}}}$$

## RESULT AND DISCUSSION

The extracted topic phrases will be used as cluster labels to represent meaningful document topics and the clusters will yield the user simple interface which is used to provide the user search for topic contents. The clustering module will take the user query and automatically feed it to a search engine like Google using its Application Programmable Interface and the user will be presented with the final cluster structure, each cluster having attached to it a topic phrase label and a description, and inside the cluster each web page having its own description and link to the page. In Figure 2, we have represented the place of the cluster topic phrase label for cluster description. In the result of figure 2, we have represented the output returned by the intelligent topic searcher (eFactfinder) as a result of user query "What is a jaguar", showing in detail the topic representation.

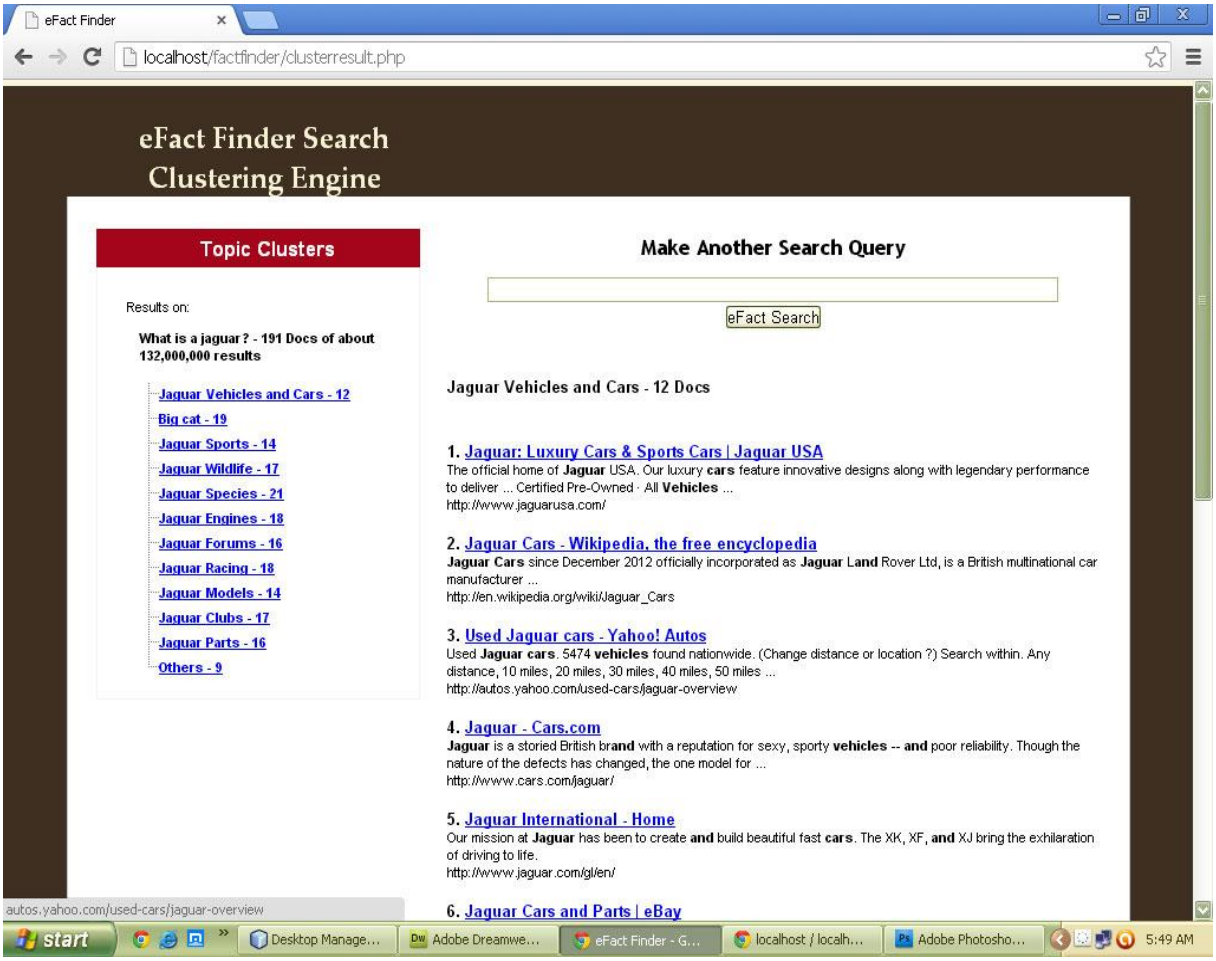


**Fig. 2. Detailed representation of Topic Phrase Cluster Labels: application output**

When we want information or help from a person, we use words to make a request or describe a problem, and the person replies with words. Unfortunately, computers do not understand human language, so we are forced to use artificial languages and unnatural user interfaces. In science fiction, we dream of computers that understand human language, that can listen to us and talk with us. To achieve the full potential of computers, we must enable them to understand the semantics of natural language.

This paper proposed a novel task and also a set of initial techniques for finding and compiling topic specific knowledge (concepts and definitions) on the Web. The proposed techniques aim at helping Web users to learn an unfamiliar topic in-depth and systematically. Given a search query, the system first discovers salient concepts of the topic from the snippets returned by the search engine. It then identifies those informative pages containing definitions of the search query on the topic and its salient concepts. Due to the convenience of the Web along with its richness and diversity of information sources, more and more people are using it for serious learning. It is important that effective and efficient systems be built to discover and to organize knowledge on the Web, in a way similar to a traditional book, to assist learning.

## REFERENCES

**Ashish, N. and Knoblock, C. (1997).**Wrapper generation for semi structured Internet sources.. SIGMOD Record, 26(4), 1997.

**Barker, K., Cornacchia, N. (2000):** Using noun phrase heads to extract document keyphrases.In Proc. of the thirteenth Canadian Conference on Artificial Intelligence, pages 40-52, 2000.

**Bassma S. Alsulami, Maysoon F. Abulkhair, Fathy A. Essa (2012).** "Semantic Clustering Approach Based Multi-agent System for Information Retrieval on Web". International Journal of Computer Science and Network Security, Vol. 12 No. 1

**Brin, S. and Page, L. (1998).** The anatomy of a large-scale hypertextual web search engine.. WWW7, 1998.

**Ceri, S., Comai, S., Damiani, E., Fraternali, P., and Tranca, L. (2000).** Complex queries in XML-GL.. In SAC (2) 2000:888-893.

**Chengzhi Zhang, Huilin Wang, Yao Liu, Dan Wu, Yi Liao, Bo Wang. (2008).** Automatic Keyword Extraction from Documents Using Conditional Random Fields. Journal of Computational Information Systems, 2008

**Cohen, W., Fan, W. (1999).** Learning page-independent heuristics for extracting data from Web pages.. WWW8, 1999.

**Cohen, J. D. (1995).** Language and domain-independent automatic indexing terms for abstracting. Journal of the American Society for Information Science, 1995

**Croft .W .B (1987).** Approaches to intelligent information retrieval. Information Processing and Magement, 23:249-254, 1987.

**Dean, J. and Henzinger, M.R. (1999).** Finding related pages in the World Wide Web.. WWW8, 1999.

*Scientia Africana, Vol. 13 (No.2), December 2014. Pp53-68*
*© College of Natural and Applied Sciences, University of Port Harcourt, Printed in Nigeria* *ISSN 1118 - 1931*

**Feldman, R., Liberzon, Y., Rosenfeld, B, Schler, J. and Stoppi, .J .A (2000).** framework for specifying explicit bias for revision of approximate information extraction rules.. KDD-00, 2000.

**Guan, T. and Wong, K.F.(1999).** KPS . a Web information mining algorithm.. WWW8, 1999.

**[Giunchiglia, F., Shvaiko, P., Yatskevich, M. (2004).:** S-Match: An algorithm and an implemented of Semantic Matching. In Proc. Of ESWS' 2004.

**Helen .J Peat and Peter Willett (1991).** The limitation of term co-occurence data for query expansion in document retrieval systems. Journal of the American Society for Information Science, 42(5):378-383, 1991.

**Hulth, A (2003).** Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, Sapporo, Japan, 2003

**Jianga, X. (2009).** A ranking approach to keyphrase extraction, Microsoft Research Technical report (MRT'09)

**Keith Humphreys. J. B. (2002).** Phraserate: An HTML keyphrase extractor. Technical Report. 2002

**Kleinberg, J. (1998).** Authoritative Sources in a Hyperlinked Environment.. Proc. of ACM-SIAM Symposium on Discrete Algorithms, 1998.

**Kumar, S., Raghavan, P., Rajagopalan, S., Tomkins, A. (1999).** Extracting large-scale knowledge bases from the Web.. VLDB-99, 1999.

**Liu, F.; Liu, Y. (2008).** Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion. In proceedings of the University of Texas at Dallas, Institute of electrical and electronics engineers (IEEE)

**Lu .X (1990).** Document retrieval: A structure approach. Information Processing and Magement, 26(2):209-218, 1990.

**Matsuo, Y, Ishizuka, M. (2004).** Keyword extraction from a single document using word co-ocuurrence statistical information. International Journal on Artificial Intelligence Tools, 2004

**Maron, .M .E and Kuhns .J .K (1960).** On relevance, probabilistic indexing and information retrieval. Journal of the ACM, 7:216-244, 1960.

**Mendelzon, A., Mihaila, G. and Milo, T. (1997).** Querying the World Wide Web.. Journal of Digital Libraries 1(1): 68-88, 1997.

**Miller .G (1990).** Wordnet: an on-line lexical database. International Journal of Lexicography, 3, 1990.

**Miller, J. (1990).** Wordnet: An online lexical database. International Journal of Lexicography,Vol.3(4).

**Ngu, D.S.W. and Wu, X (1997).** Site Helper: A localized agent that helps incremental exploration of the World Wide Web.. WWW6, 1997.

**Nie .J. Y and Brisebois .M. (1996).** An inferential approach to information retrieval and its implementation using a manual thesaurus. Artificial Intelligence Review, 10:1-31, 1996.

**Norbert Fuhr (1992).** Probabilistic models in information retrieval. The Computer Journal, 35(3):243-255, 1992.

**Ogawa, Y. (1993).** Simple word strings as compound keywords: An indexing and ranking method for Japanese texts. Proceedings of 16th annual international ACM-SIGIR Conference on Research and development in information retrieval.

**Plas, L. Pallotta, V. Rajman, M., Ghorbel, H. (2004).** Automatic keyword extraction from spoken text. A comparison of two lexical resources: the EDR and WordNet. Proceedings of the 4th International Language Resources and Evaluation, European Language Resource Association, 2004

**Robertson .S, Maron .M, and Cooper .W (1982).** Probability of relevance: a unification of two competing models for document retrieval. Information Technology: Research and Development, 1:1-21, 1982.

**Sinclair .J (1991).** Corpus, concordance, collocation. Oxford University Press, 1991.

**Song, M., Song, I., Hu, X.(2003).** Kpsptter: A flexible information gain-based keyphrase extraction system. In Proc. of the fifth ACM international workshop on web information and data management, pages 50-53.ACM press, 2003

**Quinlan, J.(1996).** Learning decision tree classifier. ACM Computer Survey. 28(1):71-72, 1996.

**Tadeusz Radecki (1979).** Fuzzy set theoretical approach to document retrieval. Information Processing and Magement, 15:247-259, 1979.

**Turney, P. (2000).** Learning Algorithms for keyphrase extraction. Information retrieval-INRT National research council, Vol.2, No.4,303-336.

**Turtle .H and Croft W. B (1990).** Inference network for document retrieval. In Proceedings of 13th ACM-SIGIR Conference, Brussels, 1990.

**Underwood, G. Maglio, P. and Barrett, R.(1998).** User-centered push for timely information delivery.. WWW7, 1998.

**Van Rijs Bergen .C .J (1986).** A non-classical logic for information retrieval. The Computer Journal, 29(6):481-485, 1986.

**Van Rijsbergen . C. J (1979).** Information Retrieval. Butterworths, London, 2 edition, 1979.

**Witten, I., Paynte, G,. Frank, E., Gutwin, C., Nevill-Manning, C. (1999).** KEA: practical automatic keyphrase extraction. In Proceedings of the 4th ACM Conference on Digital Library, 1999

**Xinghua u and Bin Wu, (2006).** "Automatic Keyword Extraction Using Linguistics Features ", Sixth IEEE International Conference on Data Mining(ICDMW'06), 2006.

**Zhang, C. (2008).** Automatic keyword extraction from documents using conditional random fields. Journal of computational and information systems 4:3,1169-1180.