A Refined Methodology for Automatic Keyphrase Assignment to Digital Documents

Sharifullah Khan, Iram Fatima, Rabia Irfan, Khalid Latif
School of Electrical Engineering & Computer Science
National University of Sciences and Technology (NUST)
H-12 Islamabad, Pakistan
{sharifullah.khan, iram.fatima, rabia.irfan, Khalid.latif}@seecs.edu.pk

Journal of Digital Information Management

ABSTRACT: *Keyphrases precisely express the primary topics and themes of documents and are valuable for cataloging and classification. Manually assigning keyphrases to existing documents is a tedious task; therefore, automatic keyphrase generation has been extensively used to classify digital documents. Existing automatic keyphrase generation algorithms are limited in assigning semantically relevant keyphrases to documents. In this paper we have proposed a methodology to refine the result set of automatically generated keyphrases by Keyphrase Extraction Algorithm (KEA++), so that the keyphrases accurately and precisely represent the content of the document. Our approach is an additional layer at the top of KEA++ and exploits semantic relationships and hierarchical structure of the controlled vocabulary to filter out irrelevant keyphrases from the result set generated by KEA++. The methodology was applied on different sets of academic publications for evaluation. Evaluation demonstrates that the proposed refinement methodology improves the quality of generated keyphrases.*

Categories and Subject Descriptors
I.7.2 [Document Preparation]; Index generation: **H.3.1 [Content Analysis and Indexing];** Indexing methods

General Terms: Digital documents, Key words, Indexing

Keywords: Keyphrase assignment, Automatic indexing, Vocabulary

Received: 11 October 2010; Revised 19 November 2010, Accepted 10 December 2010

1. Introduction

Keyphrases express the primary topics and themes of a document precisely [7, 8, 13, 16]. A keyphrase is defined as meaningful and significant expression, that describes the content of the document accurately and precisely, consisting of single word, e.g. information, or multi-word compound terms, e.g. information retrieval. They are widely used for different applications such as text clustering and classification, content-based retrieval, automatic text summarization, thesaurus construction, searching and navigation. Mostly documents in digital libraries are submitted without keyphrases, especially in the case of journal and conference articles. Manual generation of keyphrases for digital documents is complex and time consuming. Therefore it is beneficial to automatically generate keyphrases for documents to represent their main contents.

Existing approaches for keyphrase generation are generally classified into *keyphrase extraction and keyphrase assignment* [6, 14]. In keyphrase extraction, the phrases occurring in a document are analyzed to identify apparently significant ones, on the basis of properties such as frequency and length. In contrast to extraction, keyphrase assignment is used to generate keyphrases from controlled vocabulary (a.k.a. taxonomy) and documents are aligned according to their contents that correspond to the elements of controlled vocabulary. Some of the existing automatic tools perform only keyphrase extraction; others can perform only keyphrase assignment. The existing keyphrase assignment algorithms generate many irrelevant keyphrases along with relevant ones. The quality of the generated keyphrases by the existing keyphrase assignment approaches has not been able to meet the required accuracy level of applications [2, 14].

One of the most famous automatic keyphrase generation tools is Keyphrase Extraction Algorithm (KEA++). KEA++ uses the hybrid approach i.e. it performs both keyphrase extraction and keyphrase assignment. It uses Naive Bayes classifier [12, 10, 11] for automatic keyphrase generation of digital documents. KEA++ is good at identifying relevant keyphrases using linear controlled vocabulary but spurs a lot of irrelevant keyphrase in results for hierarchical controlled vocabulary. The main focus of this research is to improve the quality of keyphrases generated through automatic keyphrase assignment. Our approach is an additional layer at the top of KEA++. Our objective in this research is to filter out irrelevant keyphrases generated by KEA++. Our methodology refines the result of keyphrase assignment to documents by exploiting dif- ferent hierarchical levels of controlled vocabulary. It identifies the keyphrases that are more close to human intuition as compared to KEA++.

The rest of the paper is organized as follows. Section 2 explains controlled vocabulary. Related work has been discussed in Section 3. Section 4 explains our proposed refinement methodology of automatically generated keyphrases. Section 5 makes clear the methodology by walk-through examples considering different scenarios. In Section 6, we evaluate the proposed methodology and conclude the paper in Section 7. It also provides future directions where the research work can be extended.

2. Controlled Vocabulary

Controlled vocabulary also known as taxonomy or classification scheme, can be described as the formal specification of the concepts belonging to certain domain and the relationship amongst them. As part of this research, we assume that controlled vocabulary, taxonomy and classification scheme are the same, however in actual there exist some differences between them. In order to better understand the remaining

C. Computer System Organization (First Level)
C.2 Computer Communication Networks (Second Level)
C.2.1 Network Architecture and Design (Third Level)
C.2.1.0 Asynchronous Transfer Mode (ATM) (Forth Level)
C.2.1.1 Circuit Switching networks (Forth Level)
C.2.1.2 Network Communication (Forth Level)
C.2.1.3 Network Topology (Forth Level)
C.2.1.4 Packet-switching networks (Forth Level)
C.2.1.5 Store and forward networks (Forth Level)
C.2.1.6 Wireless Communication (Forth Level)
C.2.2 Network Protocols (Third Level)
C.2.2.0 Applications (SMPT, FTP, etc) (Forth Level)
C.2.2.1 Protocol Architecture (OSI Model) (Forth Level)
C.2.2.2 Protocol Verification (Forth Level)
C.2.2.3 Routing Protocols (Forth Level)

sections it is better to have a look at the structure of taxonomy. The following script is a part of the ACM Computing Classification Scheme[1], which is the taxonomy used for computing domain.

Each and every keyphrase in the above script represents a concept. The taxonomy contains a set of 1287 concepts (i.e. topics) in the Computer Science domain and relations between them. It has a 4 levels tree (i.e. containing three coded levels and a fourth uncoded level) and 16 separate concepts called eneral Terms that are applied to all areas, languages, theory and human factors. Each keyphrase is identified by alphanumeric code (i.e. label), such as C.2 or C.2.2. The labels are meaningful in a sense that one can judge from the labels the hierarchical level of the keyphrase in the classification scheme, for instance, C.2 shows that the keyphrase "*Computer Communication Networks*" is at the second level of the taxonomy. In ACM Computing Classification the lowest hierarchical level is four and keyphrases at the lowest level are uncoded. However, we have assigned labels (i.e. codes) to keyphrases at fourth level using the ACM logic for this research purpose. For instance C.2.2.0, C.2.2.1 are self assigned labels and are not available in ACM Computing Classification.

Each keyphrase (i.e., concept) has some sub-keyphrases (i.e. sub-concepts) which are referred as narrower keyphrases such as "*Network Architecture and Design*" (C.2.1) has a sub concept "*Asynchronous Transfer Mode (ATM)*" (C.2.1.0). Similarly a sub-concept has some broader concepts, such as "*Computer Communi- cation Networks*" (C.2) is broader keyphrase of "*Network Architecture and Design*" (C.2.1). The organization of keyphrases in broader and narrower levels forms the hierarchical structure of taxonomy. Moreover, some keyphrases have some semantically equivalent keyphrases in ACM Computing Classification, for example, "*Control Structures and Microprogramming*" (B.1) is equivalent to "*Language Classifications*" (D.3.2). In other words we can say that these concepts are semantically similar or related to each other.

In actual classification usage, first-level nodes (like B. Hardware) are never used to classify material. For material at a general level, the General node (in this case B.0) is used instead. The General node at the first or second level can serve two purposes: it is used for papers that include broad treatments of the topic covered by its parent node (the node immediately preceding it in the tree), or it may cover several topics related to some (but not necessarily all) of its sibling nodes. For example, under K.7 "*Computing Profession*", node K.7.0 General would be used to classify a general article on the *Computing Profession*, but also could be used for an article that dealt specifically with *Computing Occupations* (K.7.1), *Organizations* (K.7.2) and *Testing, Certification*, *and Licensing* (K.7.3). A language that is being widely used to represent taxonomies is SKOS[2]. A SKOS snippet of the taxonomy in Turtle syntax showing "C.2.3 *Network Operations*" is presented in the listing below.

```
<http://www.acm.org/class/C.2.3>
rdf:type          skos:Concept;
skos:broader      acm:C.2 ;
skos:inScheme     acm:Computing Classification ;
skos:narrower     acm:C.2.3.0,
                      acm:C.2.3.2 ,
                      acm:C.2.3.1 ;
skos:prefLabel “Network Operations”@en .
```

3. Related Work

Both keyphrase generation techniques: keyphrase extraction and keyphrase assignment yield almost similar level of accuracy and have their advantages and limitations. Many techniques have been developed to perform the task of automatic keyphrase generation but we are particularly interested in the machine learning techniques. Machine learning techniques need two sets of documents, one for training purpose and other for evaluation of the model. A statistical model is learned after analyzing feature values of each candidate and this learning process depends on the training scheme of the algorithm [12, 10]. The existing tools like KEA [3, 19], GenEx [17, 18], and Hulth's approach [5] adopt machine learning techniques.

In KEA [3, 19], keyphrase extraction can be achieved by performing two main steps: (a) candidate selection and (b) filtering. In the first step, all compound terms, excluding stop-words are extracted from documents as candidates. In the second step, it analyzes the selected candidate keyphrases. Candidate keyphrases consist of one word or concatenation of two or more words (tokens) that do not begin or end with a stop-word [3, 19]. A Naive Bayes Learning scheme is used to create a statistical model from training data. In filtering, for each candidate keyphrase, KEA uses (a) keyphrase frequency and (b) distance of the keyphrase's first occurrence in the document from its beginning. Then it calculates the overall probability for each candidate keyphrase in order to rank it.

GenEx [17, 18] keyphrase extraction algorithm has two main components (a) Genitor and (b) Extractor. Genitor is applied to determine the best parameters setting from the training data. Extractor combines a set of symbolic heuristics to create a ranked list of keyphrases. Hulth's algorithm [5] also uses natural language processing (NLP) techniques in addition to machine learning for keyphrase extraction. Candidates are filtered on the basis of four features (a) term frequency, (b) inverse document frequency, (c) position of the first occurrence, and (d) part of speech tagging. Hulth's evaluation results are slightly higher than those reported for KEA and GenEx. Hulth's observations are good motivation to explore further NLP techniques for important keyphrase extraction and assignment. GenEx is based on very complex heuristics for filtering but it does not outperform KEA so KEA is the simplest keyphrase extraction approach among these systems.

[1]http://www.acm.org/about/class/1998/ [Feb 18, 2011]

[2]http://www.w3.org/2004/02/skos/ [Feb 18, 2011]

In keyphrase assignment, controlled vocabulary is used that describes the characteristics of knowledge source in order to find semantically relevant keyphrases [4, 12, 10, 11]. The task of keyphrase assignment is similar to text classification or categorization [15]. Methods of automatic text classification have been developed for around fifty years. Until late 80's first logical text classification rules were created manually and then applied on electronic documents [9]. In early 90's machine learning and different inductive learning schemes have been applied to analyze the manually classified documents and build the classifier [15].

As stated earlier, KEA++ [12, 10, 11] is a hybrid approach, i.e. it performs both keyphrase extraction and keyphrase assignment . It is based on a machine learning technique and uses the Naive Bayes statistical model to train the model and to extract keyphrases. It involves taxonomy in extracting semantically equivalent keyphrases from documents. KEA++ needs to be trained on a set of documents along with their controlled vocabulary before extracting unknown semantic keyphrases from documents. KEA++ takes a document along with the controlled vocabulary as input for keyphrase extraction. KEA++ returns those keyphrases of controlled vocabulary to which the document is semantically aligned. But the results of KEA++ still contain noise. In order to filter out the irrelevant information from the generated keyphrases of KEA++ there is a need for a refinement methodology that reduces the noise in the result set of KEA++.

4. Proposed Methodology

The focus of this research is the refinement process to improve the quality of generated keyphrases. The proposed methodology refines the result set of keyphrases returned by KEA++ [12, 10, 11] using ACM Classification scheme. The methodology comprises three sub processes: (a) parameters setting of KEA++, (b) refinement rules, and (c) refinement algorithm. refinement rules are applied on the set of keyphrases returned by KEA++ according to proposed refinement algorithm and return the most relevant keyphrases and discard irrelevant keyphrase from KEA++ result set.

4.1 Parameters setting of KEA++

KEA++ can be applied on different data sets with customized parameters setting in order to generate the most relevant keyphrases. Parameters setting of KEA++ depend on taxonomy and documents' length. The statistical model should be trained on the optimum hierarchical level of the taxonomy. Training of KEA++ on top levels of hierarchy in the taxonomy affects the accuracy of the results. We utilized ACM Computing Classification Scheme as controlled vocabulary in SKOS format using UTF-8 encoding. Our proposed strategy used for the customization of KEA++ parameters setting in the refinement methodology is as follows.

1. Maximum Length of keyphrases: Five words. After analyzing the ACM Computing Classification scheme, we set the value of this parameter to five words. This value covers the common maximum available length of keyphrases in taxonomy that can be associated with the documents.
2. Minimum Length of keyphrase: Two words. Minimum keyphrases length is one word in ACM Computing Classification scheme which are at the top level of the taxonomy. The top level keyphrases are very general ones and generally not assigned to documents. We set the value of this parameter to two words because setting the value to one word provides many irrelevant keyphrases.
3. Minimum frequency of keyphrase: Two times. KEA++ recommends two words for this parameter in lengthy documents. If the parameter value is less than two words in a document, then KEA++ returns many irrelevant keyphrases. It returns very few keyphrases if the value of the parameter is greater than two words and may neglect relevant keyphrases.
4. Number of Extracted Keyphrases: Ten words. If the value to this parameter is less than ten words, for example four words, and then KEA++ returns the first four keyphrases from the results it generates. These keyphrases might not be relevant. Other parameters setting can affect the result of this parameter as mentioned in the above paragraphs.

4.2 Refinement rules

We have designed refinement rules after the deep analysis of the working behavior of the KEA++. These rules emphasize the importance of different hierarchical levels of the taxonomy in the training and keyphrase generation process. The main contribution of this research is to discard irrelevant keyphrases from the results of KEA++ by considering the hierarchical structure of taxonomy. Following are our proposed refinement rules.

- Rule I: Adopting training-level. By training-level we mean the hierarchical level of taxonomy which has been adopted in manually generated keyphrases for documents in the training data set of KEA++. In other words, if manual keyphrases in training data set are mostly aligned at the third level of the taxonomy then the training-level in the data set is three (03). This rule proposes to adopt the training-level of taxonomy for refining the KEA++ keyphrases result set. The effective usage of the remaining rules depends on the accurate value of the training-level of taxonomy.
- Rule II: Retaining training-level keyphrases. This rule intends to retain those keyphrases in the refined result set from KEA++ result set that are aligned on the training-level of taxonomy. For instance, if manual keyphrases in the training data set are mostly aligned at the third level of the taxonomy then we shall retain all those keyphrases in the refined result set from the KEA++ result set that are aligned at the third level of the taxonomy.
- Rule III: Stemming keyphrase narrower than training-level and aligned at General Node. If a keyphrase is narrower than the training-level and aligned on General node (i.e., explained in Section 2); then we stem the narrower keyphrase to its training-level keyphrases. For example:

A. General Literature (First level)
  A.0 General (Second level)
    A.0.0 Biographies/autobiographies (Third Level)
    A.0.1 Conference proceedings (Third Level)
    A.0.2 General literary works (e.g., fiction, plays) (Third Level)

Assume that the training-level is the Second Level. If the KEA++ result set contains narrower keyphrase that is aligned on the General node, such as, Biographies/autobiographies (A.0.0), so it will be stemmed to its broader training-level, i.e. A.0.

- Rule IV: Preserving keyphrases narrower than training-level. We preserve keyphrases narrower than training-level in the KEA++ result set if the result set does not contain training-level keyphrases. This rule identifies the relevant keyphrases in the absence of training-level keyphrases in the KEA++ result set.

- Rule V: Identifying and preserving keyphrase Equivalent to training-level keyphrase. In KEA++ result set, if keyphrases broader than training-level and have Equivalent training-level keyphrases, then replace the broader keyphrases with their respective Equivalent training-level keyphrases, and preserve them in the refined result set.
- Rule VI: Removing redundant keyphrase (KP). It discards redundant keyphrases from the refined result set of keyphrases.

4.3 Refinement algorithm

In this section we describe the proposed refinement algorithm for the implementation of the proposed refinement rules. Algorithm-1 describes the proposed algorithm steps that are performed to get the refined result set of keyphrases from the KEA++ result set. First of all parameters of the KEA++ are customized and then trained on the set of documents using taxonomy. Adopting the training-level for the refinement rules has primary importance because it guides the remaining refinement rules. Then KEA++ generates keyphrases for test documents (test data). The keyphrases returned in KEA++ result set are processed to get their level labels from the taxonomy. Identifying level labels is required before applying the refinement rules because they represent the hierarchical order of the keyphrases.

If the KEA++ result set has training-level Keyphrases then these training-level keyphrases are retained in the refined result set. Narrower than training-level keyphrases in the KEA++ result set are stemmed to their respective training-level keyphrases and kept in the refined result set if they are aligned on General node in taxonomy; otherwise narrower level keyphrases are discarded. Broader than training-level keyphrases in the KEA++ result set are replaced with their respective Equivalent training-level keyphrases, if they have ones in taxonomy, and preserved them in the refined result set.

If the KEA++ result set does not contain any training-level keyphrase then the narrower than training-level keyphrases in the KEA++ result set are preserved in the refined result set. Broader than training-level keyphrases in the KEA++ result set are replaced with their respective Equivalent training-level keyphrases, if they have ones in taxonomy, and preserved in the refined result set. Finally redundant keyphrases are discarded from the refined result set of keyphrases.

5. Walk-through Examples

The following two examples explain the proposed refinement methodology with two different cases of the refinement algorithm.

Title	Passive Estimation of Quality of Experience
Identification Key	JUCS, Vol. 14, Issue 5, year 2008
Manual Keyphrases	C.2.3 (Network Operations), C.4 (Performance of Systems)

Table 1. Sample Document Metadata for Example 1

Algorithm 1 Keyphrase refinement in KEA++ Result Set

1. Customize the parameters of the KEA++.
2. Train the KEA++ on documents and taxonomy.
3. Generate keyphrase result set with KEA++ for unknown documents.
4. Adopt the training-level from the training data set.
5. Identify the labels of keyphrases from taxonomy in KEA++ result set.
6. Initialize refined result set.
7. **If** KEA++ result set contain (Levels of keyphrases are narrower OR broader than training-level) AND contain (Levels of keyphrases are equivalent to training-level) **then**
   (a) **If** (Levels of keyphrases are equivalent to training-level) then preserve training-level keyphrases in Refined result set
   (b) **Else If** (Levels of keyphrases are narrower OR broader than training-level) **then**
      i. **If** (Levels of Keyphrases are broader than training-level) **then** identify and preserve their Equivalent training-level keyphrases
      ii. **Else If** (Levels of keyphrases are narrower than training-level and aligned on General node) **then**
      stem narrower keyphrases to their respective training-level keyphrases and preserve them in Refined result set
8. **Else If** KEA++ result set contain (Levels of keyphrases are narrower OR broader than training-level) AND NOT contain (Levels of keyphrases are equivalent to training-level) **then**
   (a) **If** (Level of keyphrases are narrower than training-level) **then** preserve narrower keyphrases in Refined result set
   (b) **Else If** (Level of keyphrases are broader than training-level) **then** identify and preserve their Equivalent training-level keyphrases in Refined result set
9. Remove redundant keyphrases from the Refined result set
10. Return the Refined result set of keyphrases

KEA ++ Keyphrases	Level Labels	ReFIned Result Set
Network Management	C.2.3.0	
Distribution Functions	G.3.2	G.3.2
Network Operations	C.2.3	C.2.3
Approximate Methods	I.4.2.1	

Table 2. Keyphrase Result Set of KEA++ in Example 1

5.1 Example 1: Result set with training-level keyphrases

Table. 1 shows the title of the document, its identiFIcation key and its manual keyphrases used in this example. KEA++ generates a set of keyphrases for the document using ACM ClassiFIcation Scheme as taxonomy and our proposed customized parameters setting. Table. 2 shows keyphrases returned by KEA++. By comparing the KEA++ result set with the manual set of keyphrases for the given document, it is observed that KEA++ result set contains irrelevant keyphrases along with relevant keyphrases. Irrelevant keyphrases are not so insigniFIcant that can be ignored. The purpose of reFInement methodology is to reduce the noise in the result set of KEA++. training-level in the manual set of keyphrases is three (03) because keyphrases in the training data set are mostly aligned on third level of taxonomy.

After identifying the level labels of keyphrases from taxonomy, the refinement algorithm checks whether the level labels of keyphrases contain training-level keyphrases. As KEA++ result set contains the training- level keyphrases, i.e. C.3.2 and G.3.2, then the refinement algorithm preserves these training-level keyphrases in the refined result set. Then the algorithm checks out again the KEA++ result set for keyphrases broader than training-level, but there are no broader keyphrase in the result set. Then the algorithm checks out keyphrases that narrower than training-level and aligned on the General node in the result set. Since the KEA++ result set contains a keyphrase that is narrower than training-level and also aligned on a General node i.e. C.2.3.0, so the algorithm stems that keyphrase to its respective training-level, i.e. C.2.3 and preserve the keyphrase in the refined result set. The algorithm checks whether the refined result set has any redundant keyphrases. As the result set contains the redundant keyphrases, i.e. C.2.3 and C.2.3, so it removes the redundant keyphrases from the refined result set. Finally the refined result set contains only C.2.3 and G.3.2 i.e. "*Distribution Functions*" and "*Network Operations*" as shown in the third column of the Table. 2 .

5.2 Example 2: Result set without the training-level keyphrases

Table. 3 represents the title of the document, its identification key and its manual keyphrases used in this example. KEA++ generates a set of keyphrases for the document using ACM Classification Scheme as taxonomy and our proposed customized parameters setting. Table. 4 shows keyphrases returned by KEA++. By comparing the KEA++ result set with the manual set of keyphrases for the given document, it is observed that KEA++ result set contains irrelevant keyphrases along with relevant keyphrases. After identifying the level labels of keyphrases from taxonomy, the refinement algorithm checks whether the KEA++ result set contains training-level keyphrases. The training-level for the training data set is three (03).

Title	A Knowledge Discovery Agent for a Topology Bit-map in Ad Hoc Mobile Networks
Identication Key	JUCS, Vol. 14, Issue 7, year 2008
Manual Keyphrases	C.2.1 (Network Architecture and Design), C.2.2 (Network Protocols), C.2.3 (Network Operations)

Table 3. Sample Document Metadata for Example 2

KEA ++ Keyphrases	Level Labels	Refined Result Set
Network Topology	C.2.1.7	C.2.1.7
Routing Protocols	C.2.2.3	C.2.2.3
Information Networks	H.3.4.2	H.3.4.2
Data Structures	E.1	
Computer Applications	J	

Table 4. Keyphrase Result Set of KEA++ in Example 2

KEA++ result set does not contain the training-level keyphrases, then the algorithm checks out whether the KEA++ result set contains keyphrases broader than training-level. The KEA++ result set contains keyphrases: E.1 and J, broader than training-level. The training-level Equivalent keyphrases are searched for them in the taxonomy, but the broader keyphrases do not have Equivalent training-level keyphrase in the taxonomy, therefore they are not preserved in the refined result set and discarded.

Then the algorithm checks out keyphrases that are narrower than training-level in KEA++ result set. Since the KEA++ result set contains keyphrases that are narrower than training-level, i.e. C.2.1.7, C.2.2.3, and H.3.4.2, so the algorithm preserves the keyphrase in the refined result set. The algorithm checks whether the refined result set has any redundant keyphrases. There is no redundant keyphrase in the refined result set, so the final refined result set contains only C.2.1.7, C.2.2.3 and H.3.4.2 as shown in the third column of Table. 4.

6. Results and Evaluation

In this section, we describe the details of data set, evaluation criteria, and results of the experiments carried out to evaluate the proposed refinement methodology.

6.1 Data set specifications

The experiments were carried out with a corpus of manually annotated documents from Journal of Universal Computer Science (JUCS)[3] and ACM Computing Surveys (CompSurv)[4]. The JUCS data set consisted of 100 documents, which were mostly aligned on third level of the ACM topic hierarchy. CompSurv data set also consisted of 100 documents which were aligned mostly on the fourth level of the ACM Computing Classification. The ACM Computing Classification was used in the experiments as the taxonomy. Four experiments were performed on two data sets. The first two experiments were performed on JUCS data set and the last two experiments were performed on CompSurv data set. The JUCS first experiment was performed on 65 documents in which 50 documents were used for training and 15 were used for testing. The JUCS second experiment was performed on 100 documents in which 70 documents were used for training and 30 were used for testing. Third and fourth experiments were performed on the data set of CompSurv of 100 documents in which 70 were used for training and 30 were used for testing. The difference between them was that testing documents were different in both experiments. Documents, which were used as testing documents in the first experiment, were replaced with the training documents in the second experiment. In JUCS's documents, mostly keyphrases were aligned on the third level of the ACM taxonomy, so the training- level in JUCS data set was three (03). Similarly keyphrases of documents in CompSurv were aligned on the fourth level of the ACM taxonomy, so the training-level was four (04).

[3] http://www.jucs.org/ [Feb 18, 2011]
[4] http://www.acm.org/about/class/1998/ [Feb 18, 2011]

6.2 Evaluation criteria

Our objective in this research was to reduce noise in the Keyphrases result set generated by KEA++ through applying different heuristics and exploiting the hierarchical structure of the subject taxonomy. We evaluated our proposed refinement algorithm on the basis of the generated keyphrases by the algorithms. Keyphrase based evaluation is further divided into two categories (a) number of keyphrase generated per average number of documents and (b) quality of total generated keyphrases. The quality of total generated keyphrases is measured by the precision, recall and F-measure [10, 1] for total number of generated keyphrases. Precision, recall, and F-measure are commonplace measures in information retrieval. In the quality evaluation, the manually generated keyphrases can be used as the gold standard to assess the quality of automatically generated keyphrases by any algorithm. Comparing the automatically generated keyphrases with the manual generated keyphrases is shown in Table 5 that can be used to define quality measures for generated keyphrases.

	Human Indexer (manual keyphrases)		
		Relevant	**Not Relevant**
Algorithm need to be compared	**Extracted**	True Positive (TP)	False Positive(FP)
(i.e. KEA++/ refinement algorithm)	**Not Extracted**	False Negative (FN)	True Negative (TN)

Table 5. Precision and Recall Calculation Matrix

In particular, the set of automatically generated keyphrases is comprised of true positives, and false positives. False negatives are keyphrases needed but not automatically generated, while false positives are keyphrases falsely generated by the algorithm. True negatives are false keyphrases, which have also been correctly discarded by the algorithm. Intuitively, both false negatives and false positives reduce the quality of generated keyphrases.

- **Precision** can be defined as the ratio of relevant keyphrases to the number of retrieved keyphrases. It reflects the share of real keyphrases among all found ones.

$$Precision = \frac{True\ Positive}{(True\ Positive + False\ Positive)} \quad (1)$$

- **Recall** can be defined as the proportion of relevant keyphrases that are retrieved. It specifies the share of real keyphrases that is found.

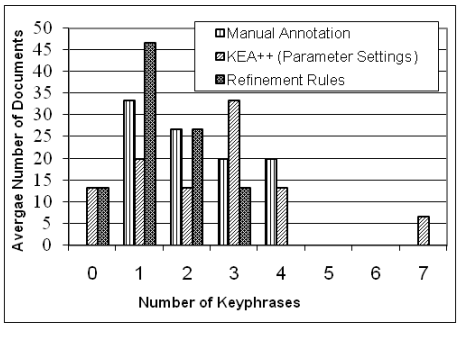
$$Recall = \frac{True\ Positive}{(True\ Positive + False\ Negative)} \quad (2)$$

- **F-measure** is defined as the harmonic mean of the Precision and the Recall. F-measure combines the Precision and Recall in a single efficiency measure.

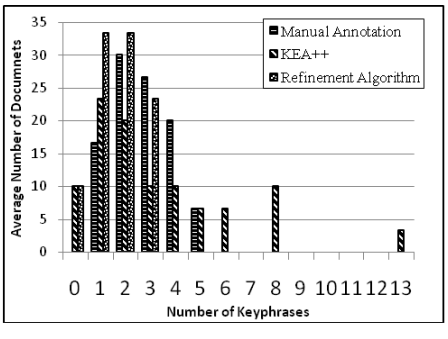
$$F - measure = 2 * \frac{(recall * precision)}{(recall + precision)} \quad (3)$$

6.3 Number of keyphrase generated per average number of documents

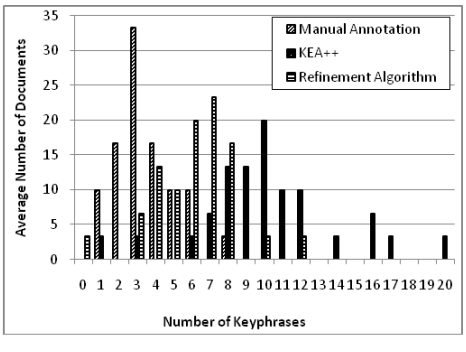
In this evaluation, we compared results among (a) manual annotation (i.e. user generated keyphrases), (b) KEA++ (i.e. with customized parameters setting) and (c) refinement algorithm in order to check noise in the result sets of keyphrase. The graphs in Figures 1 (a), (b), (c) & (d) illustrate the trend of the number of keyphrases generated per average number of documents. In the graphs, it is visible that number of keyphrases generated by the refinement algorithm is more close to manual annotation as compared to the number of keyphrases generated by KEA++. This shows that the refinement algorithm has been able to eliminate the irrelevant keyphrases from the keyphrases results set generated by KEA++.



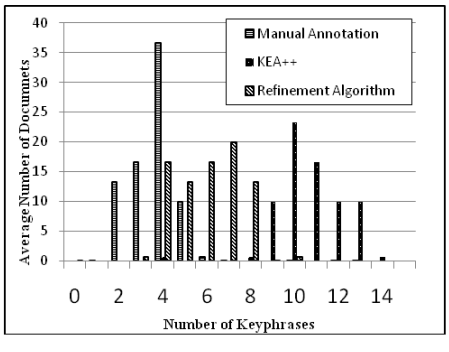
(a) JUCS Data set in Exp.1



(b) JUCS Data set in Exp.2



(c) CompSurv Data set in Exp.3



(d) CompSurv Data set in Exp.4

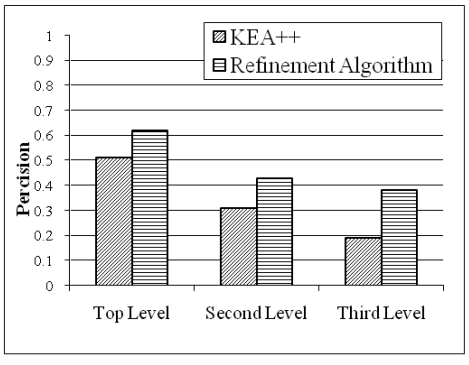
Figure 1. Number of keyphrase generated per average number of documents

6.4 Quality of total number of generated keyphrases

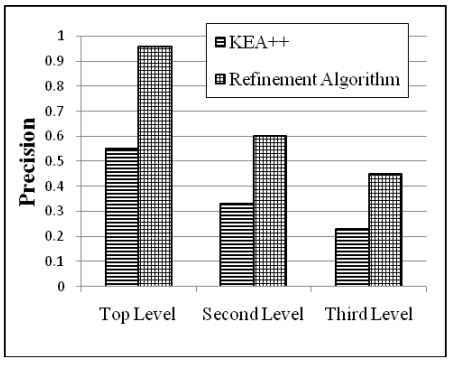
In this evaluation, we compared the KEA++ result set with the proposed refinement algorithm result set through precision, recall and F-measure of total number of generated keyphrases. Moreover in these experiments, we also illustrated the accuracy of the generated keyphrases with respect to the number of taxonomy levels, i.e. first, second and so on.

6.4.1 Precision of total number of generated keyphrases

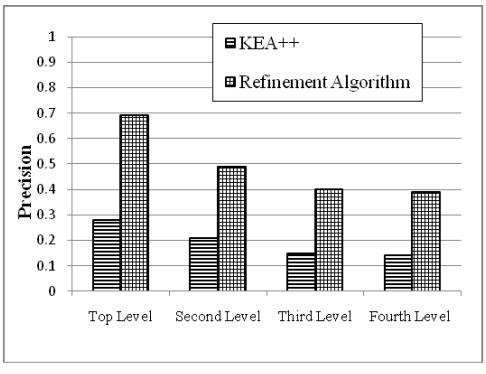
In Figure 2(a) & (b), the comparison of precision of total number of keyphrases generated by refinement algorithm and KEA++ is shown at each level of taxonomy using JUCS data set. Since the training-level in JUCS data set was three, therefore the comparison is shown on the three levels of the taxonomy. Similarly in Figure 2(c) & (d) the comparison of precision of total number of keyphrases generated by refinement algorithm and KEA++ is shown at each level of taxonomy using CompSurv data set. Since the training-level in CompSurv data set was four, therefore the comparison is shown on the four levels of the taxonomy.



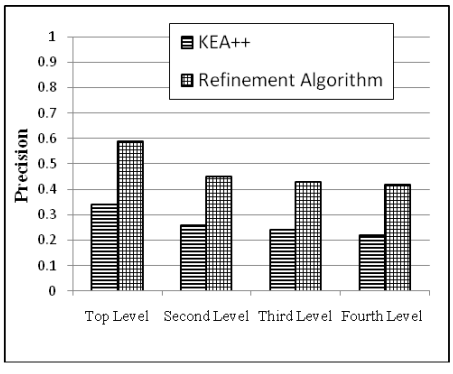
(a) JUCS Data set in Exp.1



(b) JUCS Data set in Exp.2

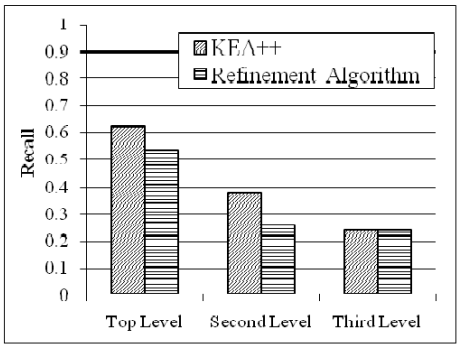


(c) CompSurv Data set in Exp.3

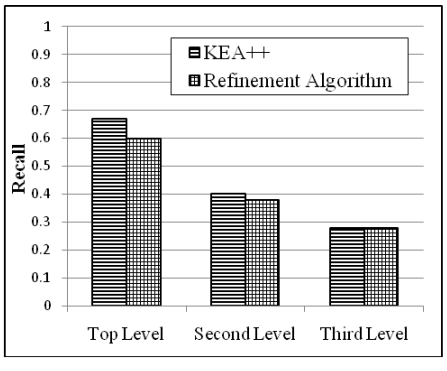


(d) CompSurv Data set in Exp.4

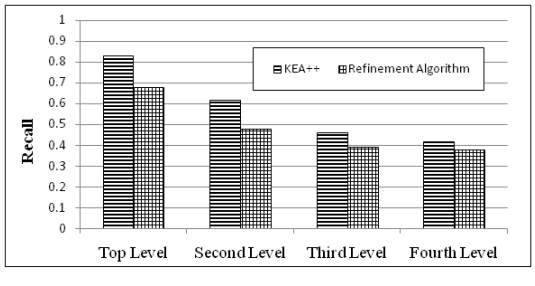
Figure 2. Precision of total number of generated keyphrases



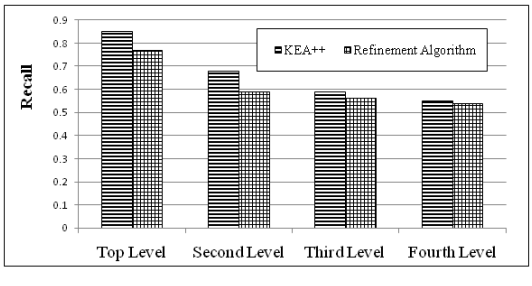
(a) JUCS Data set in Exp.1



(b) JUCS Data set in Exp.2



(c) CompSurv Data set in Exp.3



(d) CompSurv Data set in Exp.4

Figure 3. Recall against total number of generated Keyphrases

The precision of total number of keyphrases returned by refinement algorithm is higher than KEA++'s precision in each experiment on both data sets. It means that the refinement methodology discarded irrelevant keyphrases from the result set of the KEA++. Since noise in total number of keyphrases returned by refinement algorithm is reduced, so the precision is increased.

6.4.2 Recall against total number of generated keyphrases

In Figure 3 (a) & (b), the comparison of recall of total number of keyphrases returned by refinement algorithm and KEA++ is shown at each level of taxonomy using JUCS data set. Similarly in Figure 3(c) & (d) the comparison of recall of total number of keyphrases generated by refinement algorithm and KEA++ is shown at each level of taxonomy using CompSurv data set.

The recall of total number of keyphrases returned by refinement algorithm is lower than or equal to KEA++'s recall in each experiment on both data sets. The decrease in recall of total number of keyphrases returned by refinement algorithm is insignificant because the recall is remained almost same on the training- level and lower on other top levels of the taxonomy. It shows that the refinement methodology discarded irrelevant keyphrases from the result set of the KEA++.

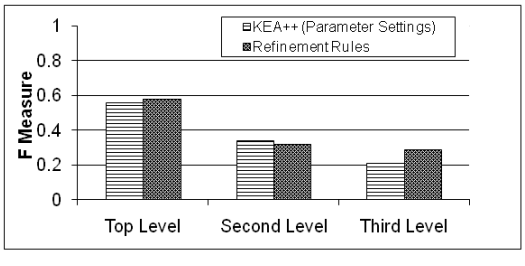
6.4.3 F-Measure against total number of generated keyphrases

In Figure 4(a) & (b), the comparison of F-measure of total number of keyphrases returned by refinement algorithm and KEA++ is shown at each level of taxonomy using JUCS data set. Similarly in Figure 4(c) & (d) the comparison of F-measure of total number of keyphrases returned by refinement algorithm is higher than KEA++'s F-measure in each experiment on both data sets. It shows that the refinement algorithm reduced noise in total number of keyphrases returned by discarding irrelevant keyphrases from the result set of the KEA++.

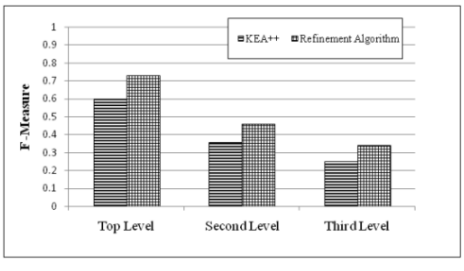
6.5 Discussion

Automatically generating keyphrases using taxonomy is by far a challenging task. None of the state-of-the- art approach has achieved high precision and recall at the same time for generating keyphrases. So, the ultimate focus was helping the user in manual classification by precisely recommending suggestions. The current systems, as reported in [2, 11, 16], have low recall as well as low precision. Our objective was to improve the precision by reducing the noise to the utmost extent possible through applying different heuristics and exploiting the hierarchical structure of the taxonomy. Our proposed algorithm decreases noise in the keyphrase result set generated by KEA++ by reducing the number of generated keyphrases per average number of documents while achieving better precision and same recall level against total number of generated keyphrases at the training-level of ACM taxonomy.

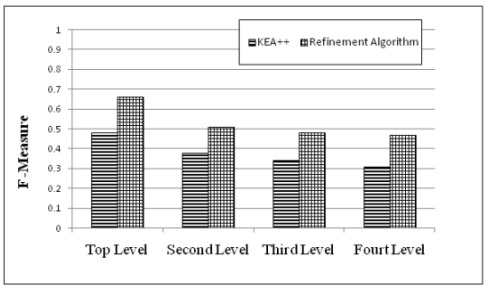
Table 6 summarizes the precision, recall and F-Measure of the proposed refinement algorithm and com- pares it with the results of KEA++ given in [11] and observed in our experiments on our selected data sets presented in the previous subsections. The precision, recall and F-Measure of KEA++ reported in [11], are 0.28, 0.26 and 0.25 respectively while the average number of manual annotation is 5.4 per document using the data set of 200 documents. Obviously precision and recall of KEA++ was affected in our result set by change in the number of documents in the data sets and average number of manual annotation per document in each data sets. In the case of the refinement algorithm, precision has been improved in all performed tests while recall is either low or equal.



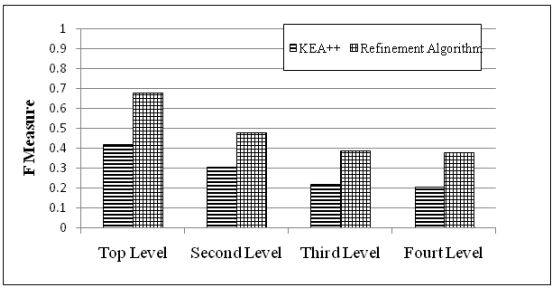
(a) JUCS Data set in Exp.1



(b) JUCS Data set in Exp.2



(c) CompSurv Data set in Exp.3



(d) CompSurv Data set in Exp.4

Figure 4. F-Measure against total number of generated Keyphrases

Date Sets	[11] Results	JUCS Data Set				ACM CompSurv Data Set			
Algorithm	KEA++	KEA++	Refnmt	KEA++	Refnmt	KEA++	Refnmt	KEA++	Refnmt
Test	Test	Test 1		Test 2		Test 3		Test 4	
Docs.	200	65		100		100		100	
Avg.# of Man. Ann.	5.4	2.27		2.35		3.46		4.5	
Precision	0.28	0.19	0.38	0.23	0.45	0.14	0.39	0.22	0.42
Recall	0.26	0.24	0.24	0.28	0.28	0.42	0.38	0.55	0.54
F-Measure	0.25	0.21	0.29	0.25	0.34	0.21	0.38	0.31	0.47

Table 6. Precision, Recall and F-Measure Statistics

7. Conclusion and Future Work

The methodology comprises three sub processes: (a) parameters setting of KEA++ and (b) refinement rules (c) refinement algorithm. The proposed refinement rules help in removing the irrelevant keyphrases from the results set generated by the KEA++. The refinement algorithm provides the functional flow to the refinement rules. The methodology exploits the hierarchical structure of the classification taxonomy. The parameters setting in the beginning of the refinement algorithm enables the KEA++ to extract the keyphrases in more optimal manner. The methodology was applied on two different data sets in four experiments. The evaluation demonstrates obvious improvement in the precision as compared to KEA++ while maintaining the same recall or low recall. Currently the focus was on a single training-level in applying the refinement algorithm for assigning the keyphrases to documents. As documents are aligned on different levels of the taxonomy in training data set, so in future this refinement algorithm can be extended to involve more than one training-levels while executing the refinement algorithm in order to achieve more accurate results. The methodology can be made more generalized by applying it in different subject domains such as Agriculture, Medicine.

References

[1] Do, H. H., Melnik, S., Rahm, E. (2003). Comparison of schema matching evaluations. *In: Web, Web Services, and Database Systems, volume 2593 of Lecture Notes in Computer Science*, p. 221 – 237. Springer Berlin Heidelberg.

[2] Dumais, S., Chen, H. (2000). Hierarchical classification of web content. *In: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* , p. 256 – 263, New York, NY, USA, July. ACM.

[3] Frank, E., Paynter, G. W.,Witten, I. H., Gutwin, C. G., Nevill-Manning, C.(1999). Domain specific keyphrase extraction, *In: Proceediings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 99)*, p. 668 – 673, San Francisco, CA, USA. Morgan Kaufmann.

[4] Gutwin, C., Paynter, G., Witten, I. H., Nevill-Manning, C., Frank, E. (1998). Improving browsing in digital libraries with keyphrase indexes. Technical report, Department of Computer Science, University of Saskatchewan, Canada.

[5] Hulth, A. (2004). *Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction*. PhD thesis, Computer and Systems Sciences, Stockholm University, Stockholm, Sweden.

[6] Jones, S., Mahoui, M.(2000). Hierarchical document clustering using automatically extracted keyphrases. *In: Proceedings of the 3rd International Asian Conference on Digital Libraries*, pages 113 – 120, Seoul, Korea.

[7] Jones, S., Paynter, G. (2002). Automatic extraction of document keyprases for use in digital libraries: evaluation and applications. *Journal of the American Society for Information Science and Technology (JASIST)*, 53 (8) 653 – 677, 2002.

[8] Jones, S., Paynter, G. W (2003). An evaluation of document keyphrase sets. *Journal of Digital Information*, 4 (1).

[9] Markey, K. (1984). Inter-indexer consistency test: A literature review and report of a test of consistency in indexing visual materials, *Library and Information Science Research: An International Journal*, 6 (2) 155 – 77, June.

[10] Medelyan, O. (2005). Automatic keyphrase indexing with a domain-specific thesaurus. Master's thesis, University of Freiburg, Germany, 2005. In English, with a German abstract.

[11] Medelyan, O., Witten, I. H.(2006). Thesaurus based automatic keyphrase indexing. *In: Proceedings of the Joint Conference on Digital Libraries*, p. 296 – 297, Chapel Hill, NC, USA.

[12] Medelyan, O., Witten, I.H. (2005). Thesaurus based automatic keyphrase indexing. *In: Proceedings of 6th Agricultural Ontology Service (AOS), workshop at EFITAWCA, Portugal*.

[13]. Paynter, G., Cunningham, S. J., Witten, I. H. (2000). Evaluating extracted phrases and extending thesauri, *In: Proceedings of the 3rd International Conference of Asian Digital Library*, p.131 – 138, Seoul, Korea.

[14] Saarti, J. (2002). Consistency of subject indexing of novels by public library professionals and patrons, *Journal of Documentation*, 58 (1) 49 – 65, January.

[15] Sebastiani, F.(2002). Machine learning in automated text categorization, *ACM Computing Survey*, 34 (1) 1 – 47, March.

[16] Thuy, D. N., MinYen, K (2007). Keyphrase extraction in scientific publications, *In: Proceedings of the International Conference on Asian Digital Libraries (ICADL)*, p. 317 – 326, Hanoi, VIETNAM.

[17] Turney, P. (1999). Learning to extract keyphrases from text. Technical report, National Research Council Canada, Ottawa, Ontario, Canada.

[18] Turney, P. (2003). Coherent keyphrase extraction via web mining. *In: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, p. 434 – 439, Acapulco, Mexico.

[19] Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., Nevill-Manning, C. G. (1999) Kea: Practical automatic keyphrase extraction. *In: Proceedings of the 4th ACM Conference on Digital Libraries* (DL 99), p. 254 – 255, Berkeley, CA, USA.