Refinement Methodology for Automatic Document Alignment using Taxonomy in Digital Libraries

Iram Fatima, Sharifullah Khan, Khalid Latif
NUST School of Electrical Engineering & Computer Science, H12 Islamabad, Pakistan
{iram.fatima, sharifullah.khan, khalid.latif}@seecs.edu.pk

Abstract

Effective information retrieval in digital libraries requires semantic alignments of documents with taxonomy. The alignments provide the semantic description of documents. The proposed methodology aligns documents using the hierarchical structure of taxonomy. It refines the results of the existing semantic keyphrase extraction algorithm. The evaluation shows promising results.

1. Introduction

Keyphrases express the primary topics and themes of a document precisely [14]. They are useful in text clustering and classification [6], content-based retrieval [1], automatic text summarization [2], thesaurus construction [12], representing search results [6], and navigation [5]. Manual assignment of keyphrases is expensive and time consuming, therefore automatic techniques are essential [4, 7]. Existing approaches for keyphrase generation are: *keyphrase extraction* and *keyphrase assignment* [13]. In keyphrase extraction, significant keyphrases in a document are identified through properties such as frequency and length. While in the latter approach, keyphrases are generated by semantically aligning a document with taxonomy. The quality of the generated keyphrases by the existing approaches has not been able to meet the required level of applications [13, 20]. Our objective is to improve the semantic alignment procedure by exploiting different hierarchical levels of taxonomy. The proposed methodology consists of a set of rules that refine the results, returned by the existing keyphrase extraction algorithm: KEA++ (Key Phrase Extraction Algorithm) [9, 10, 11]. It detects the semantic keyphrases that are more close to human intuition as compared to the previous approaches.

The rest of the paper is organized as follows. In Section 2 discusses related work. A complete workflow for methodology is described in Section 3. Section 4 covers walkthrough examples. In Section 5 results and evaluation are discussed. We conclude our findings in Section 6 and present future directions.

2. Related Work

Both keyphrase extraction and assignment may use supervised machine learning techniques. The training data are documents with manually supplied keyphrases. Keyphrase Extraction is achieved by candidate phrase identification and filtering [15]. The techniques include KEA [17], GenEx [16] and A. Hulth [6]. In KEA candidate keyphrases consist of one word or more than one word (tokens) that do not begin or end with a stop word. A Naïve Bayes based statistical models are used for training. In filtering for each candidate KEA uses (a) keyphrase frequency and (b) distance of the phrase first occurred. Then the algorithm calculates the overall probability for each candidate to rank them. GenEx [16] keyphrase extraction algorithm has two main components (a) Genitor and (b) Extractor. Genitor is applied to determine the best parameter settings from training data. Extractor combines a set of symbolic heuristics to create a ranked list of keyphrases. Hulth's algorithm [6] uses NLP tools in addition to machine learning. Candidates are filtered on the basis of four features (a) term frequency, (b) inverse document frequency, (c) position of the first occurrence (d) part of speech tag.

KEA is the simplest keyphrase extraction approach among these systems. GenEx is based on more complex filtering heuristics, but it does not outperform KEA [3]. Hulth's evaluation results are significantly higher than those reported for KEA and GenEx because of using linguistic based techniques for candidate selection and classification. Hulth's observations are a good motivation to explore further NLP techniques for keyphrase extraction and assignment.

Assignment techniques identify those keyphrases in a document that are predefined in taxonomy [3][8]. Each approach to keyphrase generation has pros and cons. Their hybrid is required to benefit from both and avoid their shortcomings KEA++: [9, 10, 11], the hybrid approach, segments each documents into individual tokens on the basis of punctuation and white spaces. KEA++ uses (a) keyphrase frequency, (b)

978-0-7695-3800-6/09 $26.00 © 2009 IEEE
DOI 10.1109/ICSC.2009.45

position of the first occurrence of the phrase, (c) length of the phrase in words, (d) level in taxonomy to determine the candidate terms. Then it applies the model built on training data using taxonomy.

In this research we refine the process of semantic keyphrase extraction from documents. The previous techniques [6, 16, 17] extract relevant information along with significant irrelevant data. The main problem here is how to separate noise from the relevant information.

3. Proposed Methodology

The proposed methodology refines the result set of keyphrases returned by KEA++ [9, 10, 11] using ACM taxonomy [18]. It comprises two major processes (a) extraction and (b) refinement. Extraction is prerequisite of refinement process. The focus of this research is the refinement process of keyphrases. Refinement process of extracted keyphrases is based on (a) parameter setting of KEA++ and (b) refinement rules. KEA++ parameters have been set according to the structure of taxonomy. Refinement rules are applied on the set of keyphrases returned by KEA++ after customized parameter settings.

3.1. Parameter Settings of KEA++:

KEA++ can be used for different data sets along with different parameter settings in order to extract the most relevant results. Parameter settings 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.

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

We set the vocabulary name to ACM computing classification in SKOS format using UTF-8 encoding. A snippet of the taxonomy in Turtle syntax showing "C.2.3 Network Operations" is presented in the listing above. Other parameters which affect the results are described below.

Max. Length of Phrases: five words. After analyzing the ACM topic hierarchy, we set the value of this parameter to five words. This value covers the common maximum available length of phrases in taxonomy that can be associated with the documents.

Min. Length of Phrase: two words. Minimum phrase length is one word in ACM taxonomy which is the top level. The top level keyphrases are very general ones and generally not associated with the extracted semantic keyphrases. We set the value of this parameter to two words because setting the value to one word provides many irrelevant keyphrases.

Min. Occurrence: two words. KEA++ recommends two words for this parameter in long documents. If the parameter value is less than two words, then KEA++ returns many irrelevant keyphrases. KEA++ returns very few keyphrases if the value of the parameter is greater than two words and may neglect relevant keyphrases.

No of Extracted Keyphrases: ten words. If the value to this parameter is less than ten words, for example four words, then KEA++ returns the first four keyphrases from the results it computes. These keyphrases might not relevant. Other parameter settings can affect the result of this parameter as mentioned in above paragraphs.

3.2. Refinement Rules

We observed in our analysis that the hierarchical levels of taxonomy and their generalization and specialization play vital role both in training and extraction process of KEA++. Refinement rules exploit the semantics of the levels and select keyphrases located on the most relevant levels. The rules are as follows.

Rule I: Adopting Training Level: The training level is the hierarchical level of taxonomy, adjusted for manually extracted keyphrases in documents and used in KEA++ training. We adopt the training level in refinement. This rule guides the remaining rules in their process.

Rule II: Preserving Training Level Keyphrases: We only preserve keyphrases aligned on the training level. This rule selects most relevant keyphrases from the KEA++ returned result set.

Rule III: Stemming Lower Level General Keyphrases: In ACM Taxonomy, there is the general category of keyphrases on each level of hierarchy. If a keyphrase is aligned on a lower level than the training level and associated with the general category in the lower level; then we stem the lower level keyprhase to its training level keyphrases.

Rule IV: Preserving Lower Level Keyphrases: If the result set of KEA++ contains no training level keyphrases, then we preserve lower level keyphrases from the result set of KEA++. This rule identifies the

relevant keyphrases in the absence of training level keyphrases.

Rule V: Identifying and Preserving Training Level Equivalent Keyphrase: Different keyphrases aligned to separate categories of ACM taxonomy can be semantically equivalent, e.g. *Control Structures and Microprogramming* (B.1) is equivalent to *Language Classifications* (D.3.2). Upper level keyphrases in the result set are replaced with their equivalent training level keyphrases, if they have any, otherwise discarded.

Rule VI: Removing Redundant Keyphrase (KP): Remove the redundant keyphrases from the refined result set.

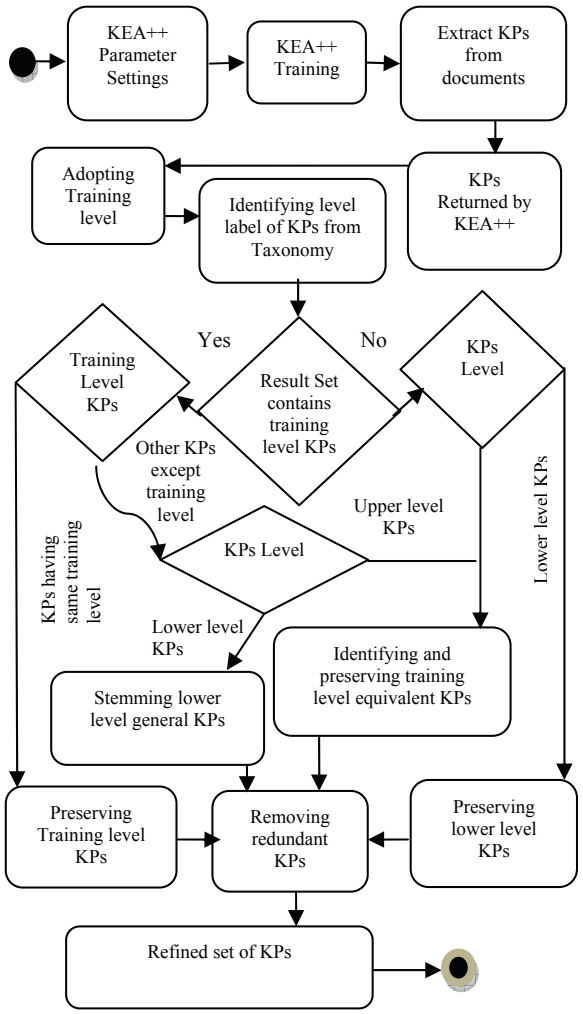

Figure.1: Keyphrase (KP) Refinement Algorithm

3.3. Refinement Algorithm

The algorithm that describes the flow of refinement rules, is illustrated in Figure 1. The first step is setting parameters of the KEA++ as mentioned in subsection 3.1. Secondly train KEA++ on the set of documents along with their keyphrases (KPs) of ACM taxonomy. Then apply KEA++ on actual documents (data).

Adopting the training level for the refinement rules has primary importance because it guides the remaining rules in their process. The keyphrases returned by KEA++ is processed to get its level label (e.g. D.3.2) in the ACM taxonomy. Indentify level labels is required before applying the refinement rules because they represent the hierarchical order of the keyphrases.

If the KEA++ result has training level keyprharse then these training level keyphrases are preserved and added in the refined result. Lower level keyphrases are stemmed to their training level keyphrases and preserved in the refined result if they are associated with the general category at the lower level in taxonomy. Otherwise lower level keyphrases in KEA++ result are discarded. Upper level keyphrases in KEA ++ result are handled according to Rule-V.

If the KEA++ result does not contain any training level keyphrases then lower level keyphrases of the result are preserved and added in the final refined result. Upper level keyphrases in KEA ++ result are handled according to Rule-V. Finally redundant keyphrases are removed from the final refined set of keyphrases.

4. Walkthrough Examples

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

4.1. Result Set with Training Level Keyphrases

Table 1 illustrates the information about a document used in the first example.

Table 1: Document Information

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

KEA++ returns the list of semantic keyphrases using ACM topic hierarchy for this document as shown in Table 2. Extracted keyphrases align the document on four keyphrases of the ACM topic hierarchy. The result set contains irrelevant keyphrases as compared with the manual annotation. The ACM taxonomy level labels of these keyphrases are shown in Table 3.

The level labels show alignment of the document on different depths in the ACM taxonomy. This result set contains the training level label i.e. C.3.2 and G.3.2. The algorithm preserves the training level keyphrases.

Moreover, the result set does not contain any upper level keyphrase, but contains lower level keyphrases. One lower level keyphrase belongs to a general category having label: C.2.3.0. So it is stemmed to the training level keyphrase, having label:C.2.3. The refined set includes redundant keyphrases, i.e. C.2.3, C.2.3, so one redundant keyphrase is discarded. After applying the refinement rules, the result set is shown in Table 4.

Table 2: Results of KEA++

Results of KEA++
Network Management
Distributed Functions
Network Operations
Approximate Methods

Table 3: Node level of KEA++ results

KEA++ Keyphrases	Level Label
Network Management	C.2.3.0
Distributed Functions	G.3.2
Network Operations	C.2.3
Approximate Methods	I.4.2.1

Table 4: Results of refinement process

KEA++ Keyphrases	Level Label	Refined Result
Network Management	C.2.3.0	
Distributed Functions	G.3.2	G.3.2
Network Operations	C.2.3	C.2.3
Approximate Methods	I.4.2.1	

4.2. Result Set without Training Level Keyphrases

Table 5 illustrates the information about a document used in the second example. Table 6 shows the refined result set. The KEA++ returned result does not contain training level keyphrases. Moreover, the result set contains both upper level and lower level keyphrases. Lower level keyphrases are preserved while upper levels keyphrases are discarded according to Rule-V.

Table 5: Sample documents

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

5. Results and Evaluation

We compare result sets of manual annotation, KEA ++ and refinement algorithm. In this comparison, third level hierarchy of the ACM taxonomy is used to ensure the precision of results. The evaluation has been performed on the basis of (a) Keyphrases and (b) documents.

Table 6: Results of refinement process

KEA++ Keyphrases	Level Label	Refined Results
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	

The dataset used in the evaluation was composed of sixty five documents taken from the Journal of Universal Computer Science (JUCS) [19], which properly follow ACM taxonomy for documents' classification. It comprises document text files and key files containing manual annotation of keyphrases that align the documents on the different levels of the ACM taxonomy. In the dataset 50 documents were used for training and 15 were used for extraction.

5.1. Evaluation based on Keyphrases

This evaluation is further divided into two categories (i) keyphrases returned per average number of documents and (ii) total returned keyphrases. In the former category we compare results among (a) manual annotation, (b) KEA++ (parameter settings) and (c) refinement rules. The graph in Figure 2 illustrates the trend of the number of keyphrases returned per average number of documents in refinement rules is more close to manual annotation. In the first category, refinement rules reduce the number of keyphrases returned against average number of documents as compared to KEA++.

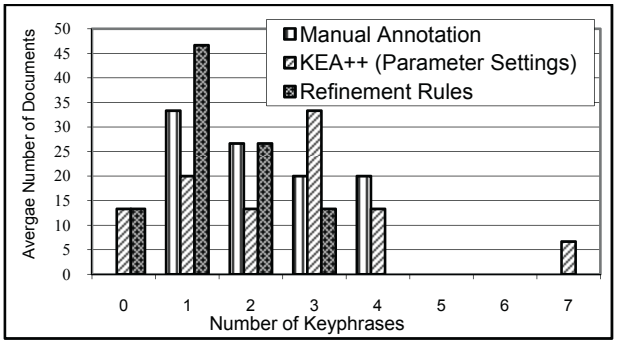

Figure 2: Keyphrases returned per average number of documents

However, it does not affect the precision of correctly aligned documents, as shown in the next subsection. The later evaluation compares the precision and recall for total returned keyphrases of both KEA++

(parameter settings) and refinement rules. Figure 3 illustrates that precision increases in the case of the refinement rules because the number of keyphrases returned per average number documents is reduced as shown in Figure 2.

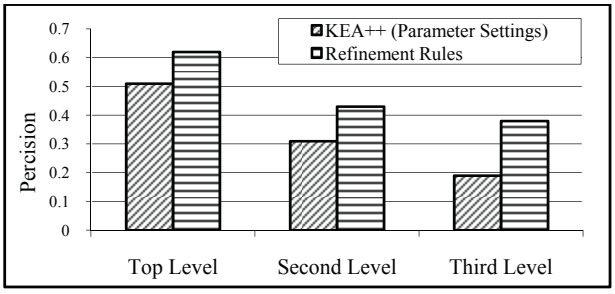

Figure 3: Precision against total keyphrases returned

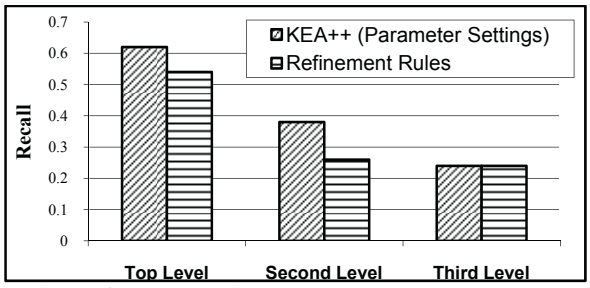

Figure 4: Recall against total keyphrases returned

Figure 4 illustrates the recall of KEA++ and refinement rules. The recall is reduced in the case of first and second level while it is the same on the third level. It shows the same performance of both approaches on the third level, i.e. the desired one.

5.2. Evaluation based on Documents

This evaluation is further categorized in (i) totally matched result and (ii) approximate matched result. The totally matched result contains all the manual annotated keyphrases of the particular document. While the approximate matched result comprises a subset of manual annotated keyphrases of the particular document. Totally matched result is more conservative approach because it ignores the approximately aligned documents. It returns inadequate searching results as compared to the approximate matched result.

Figure 5 illustrates precision for totally matched results of KEA++ (parameter setting) and refinement rules. The precision is the same in both approaches besides the refinement rules return a reduced number of keyphrases.

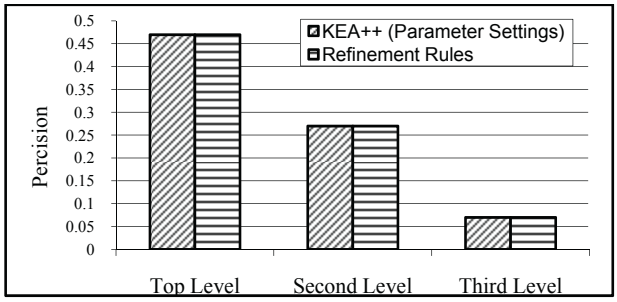

Figure 5: Precision of Totally Matched Results

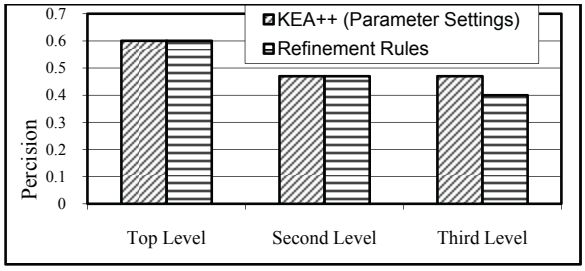

Figure 6: Precision of Approximate Matched Results

Figure 6 shows the precision of both the approaches for the approximate matched results. The precision is comparatively lower on the third level of the taxonomy in our algorithm only due to the reduced number of keyphrases per average number of documents.

5.3. Discussion

Automatically assigning digital documents to particular slots in the subject classification 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 the classification problem. So, the ultimate focus is helping the user in manual classification by precisely recommending suggestions. The current systems, as reported in [11, 14, 20], have low recall as well as low precision. Our intension here is to improve the precision by reducing the noise to the utmost extent possible through different heuristics and by exploiting the hierarchical structure of the subject taxonomy.

Our proposed methodology decreases keyphrases' noise in keyphrase extraction by reducing the number of returned keyphrases per average number of documents while achieving better precision level and the same recall level against returned keyphrases at the third level of ACM taxonomy. Moreover it maintains similar precision against the correctly aligned documents.

In [11] the precision and recall of KEA++ are 0.283 and 0.261 respectively while the average number of manual annotation is 5.4 per document in the dataset of 200 documents. While the precision and recall of KEA++ for our dataset of 65 documents (with 2.27 average number of manual annotation per document) are 0.198 and 0.24 respectively. Obviously the decrease in precision and recall is influenced by the smaller training dataset as well as comparably lower manual keyphrase assignments. The precision has been improved from 0.198 to 0.38 i.e. 191.9% on the same dataset while maintaining the same recall.

6. Conclusion and Future Work

In this paper a methodology for refinement of automatic document alignment using ACM subject classification has been introduced. The methodology takes into account semantic relations between terms that appear in the document along with different levels of the ACM taxonomy. The refinement algorithm applies the set of rules on the extracted keyphrases returned by KEA++ and refined keyphrases that are used in aligning documents with taxonomy.

An extension to KEA++ was trained and tested on documents from the Journal of Universal Computer Science (JUCS). The evaluation demonstrates that the proposed methodology significantly refines results returned by KEA++. In future, we intend to adopt multiple training levels during refinement process in order to make the methodology scalable.

7. References

[1] Arampatzis, A. T., T. Tsoris, C. H. A. Koster, and T. P. van der Weide: Phrase-based information retrieval. Information Processing and Management (1998), 34(6), 693–707.

[2] Barker, K. and N. Cornacchia: Using noun phrase heads to extract document keyphrases. In Proc. of the 13th Canadian Conference on Artificial Intelligence, (2000), pp. 40–52.

[3] Dumais, S., Platt, J., Heckerman, D.and Sahami, M.Inductive learning algorithms and representations for text categorization. Proceedings of the Seventh International Conference on Information and Knowledge Management, ACM, 1998, 148-155.

[4] Frank E., Paynter G.W., Witten I.H., Gutwin C. and Nevill-Manning C.G.: Domain-specific keyphrase extraction. Proc. 16th International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA, (1999) , pp. 668-673.

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

[6] Hulth, A. Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction. Ph. D. thesis, Computer and Systems Sciences, Stockholm University, (2004).

[7] Jones, S. and G. Paynter: Automatic extraction of document keyprases for use in digital libraries: evaluation and applications. Journal of the American Society for Information Science and Technology (JASIST) (2002), 3(8), 653–677.

[8] Leung, C.-H., and Kan, W.-K. A statistical learning approach to automatic indexing of controlled index terms. Journal of the American Society for Information Science, 1997,48, 55-66

[9] Medelyan, O: Semantically Enhanced Automatic Keyphrase Indexing. (Poster) In: Proc. of the Women in Machine Learning (WiML) Workshop co-located with the Grace Hopper Celebration of Women in Computing. San Diego, USA (2006).

[10] Medelyan, O. Automatic Keyphrase Indexing with a Domain-Specific Thesaurus. Master Thesis. University of Freiburg, Germany. (In English, with a German abstract.) (2005).

[11] Medelyan, O., Witten I. H.: Thesaurus Based Automatic Keyphrase Indexing. In Proc. of the Joint Conference on Digital Libraries 2006, Chapel Hill, NC, USA, (2006), pp. 296-297.

[12] Paynter, G., S. J. Cunningham, and I. H. Witten: Evaluating extracted phrases and extending thesauri. In Proc. of the 3rd International Conference of Asian Digital Library (2000).

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

[14] Thuy Dung Nguyen, Min-Yen Kan: Keyphrase Extraction in Scientific Publications. ICADL 2007: 317-326

[15] Turney, P.D. Coherent keyphrase extraction via Web mining, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, 2003,434-439. (NRC #46496)

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

[17] Witten I.H., Paynter G.W., Frank E., Gutwin C. and Nevill-Manning C.G. KEA: Practical automatic keyphrase extraction. Working Paper 00/5, Department of Computer Science, The University of Waikato (2000).

[18] ACM Computing Classification System, http://www.acm.org/about/class/1998/ [May 09, 2009].

[19] Journal of Universal Computer Science, http://www.jucs.org/ [May 09, 2009].

[20] Dumais, S., Chen, H. Hierarchical classification of Web content. In proceedings of 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 2000, Athens, Greece.