Madhura Marathe. Qualitative Analysis of KEA Automatic Indexing Algorithm. A Master's paper for the M.S. in I.S. degree. March, 2010. 34 pages. Advisor: José Ramón Pérez-Agüera

While working with documents on the Internet, we commonly rely on search engines to retrieve information for us. Authors of the document often identify keyphrases which would make it easier for people to search conceptually and thematically rather than solely by keywords. In the absence of author identified keyphrases, professional human indexers do this job.

With the ever growing number of documents on the Internet and with the rising popularity of digital libraries, it is almost impossible to have professional human indexers assign and manage keyphrases for all of the many documents on the Internet. For this reason, automatic indexers are built. However, accuracy and efficiency are important desirable characteristics of an automatic indexer.

This study is a qualitative evaluation of the accuracy of the KEA automatic indexing algorithm. The results of the automatic indexer are compared with the indexing done by professional indexers using a controlled vocabulary.

Headings:

Indexing

Digital libraries

Automatic Keyphrase Extraction

Controlled Vocabularies

Semantic Web

QUALITATIVE ANALYSIS OF KEA AUTOMATIC INDEXING ALGORITHM

Madhura Marathe
University of North Carolina

A Master's paper submitted to the faculty
of the School of Information and Library Science
of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements
for the degree of Master of Science in
Information Science.

Chapel Hill, North Carolina

April, 2010

Approved by:

_____

José Ramón Pérez Agüera

Table of Contents

Table of Figures

1. Introduction

Searching is something we do in everyday life. This is true even for the use of computers. Although the Internet provides an abysmal source for any queries, retrieving the exact document is still a challenge. Very often, this happens because the user enters a keyword which has the same meaning as the keyword in the document, but is not quite the same word. Most of the commonly used search engines today[1] work on the principle of directly parsing for the search string entered by the user and looking for an exact match to return the matching documents as results (Brin & Page, 1998). A useful extension to this is providing options for any non-matching word which could be possibly misspelled by the user. A common example of this is when Google search returns results for a close match saying "Did you mean…"

Evolution of libraries

Libraries are essentially a collection of written materials. Right from the thirtieth century, BC, archeologists have found writings preserved on walls, clay tablets and papyrus scrolls. Later on, such repositories came to be known as 'libraries'. Back then, libraries were private till the first public library came into being in the fourth century BC (Source: Wikipedia).

Libraries have been using indexing methods since hundreds of years. The very traditional libraries used manual indexing to group books in broader categories and then narrowing them down to more relevant categories. For example, broad categories could be Science, History, Philosophy, Arts, Literatures etc. and these categories can be further narrowed down by subdividing them into Zoology, Botany, Microbiology under Sciences and so forth. Libraries tell their users about what materials are available in that library and how to access those materials. A very basic level structuring for library collections is to label the stacks with the classification criteria. In very small community libraries, sections might be classified according to subjects and then all the books under the same subject indexed by author names or titles, much like a telephone directory.

The Renaissance period saw a considerable rise in the library collections all over the world. In the post World War II era, as the number of collections increased further, libraries started using the card catalog which is a register of all bibliographic items in a library. It is typically a cabinet of index cards that contain information about library materials and help the user in locating them in the library.

Towards the end of the twentieth century, an online version of the card catalog came into picture, called as the Online Public Access Catalog (OPAC). The OPAC is an online database containing information about the actual physical collections of a library. It serves the same functionality as a card catalog, except that it is online instead of in a cabinet.

Nowadays, digital libraries are getting more and more common with the advent of the Internet and given the convenience of using online materials. A digital library is an online collection of materials in digital form instead of having physical prints. This collection can be accessed by the local computer or remote computers, via a network, typically the Internet. In digital libraries, users are guided by controlled vocabularies so as to make sure that the entered keywords are 'known' to the search engine. This is done by feeding in the most likely search phrases to the database against which the entered search string will be matched. These key phrases are appropriately categorized by professional human indexers after careful consideration.

However, given the constantly growing number and variety of electronic documents on the web, it is impossible even for professional human indexers to index all of these documents. For this reason, the demand for automatic indexers is growing. The automatic indexing algorithms available currently do achieve the goal of automatic indexing to some extent, but are not entirely efficient for the entire repository of documents on the web and are hence, not entirely practical.

Folksonomy and Social Tagging

With the rise of social networking sites such as Facebook, MySpace, Flickr etc., people are becoming more aware of folksonomies. Web 2.0 introduced the social networking world to people. Using social tagging enabled users to use folksonomies without really understanding the underlying working of controlled vocabularies. According to Vander Wal, folksonomy is a result of personal tagging of

online objects (which can be identified by a URL) for one's own retrieval. Tagging is done in a social environment. These days we come across tags when people are tagged in photographs, stories, videos, news etc. This makes it easier to retrieve information later by just running a query which perform an action: Retrieve all photos of Paul from his DC trip. All these results of queries make use of the information retrieval system of matching keywords such as name of the person, place, date range, types of files etc.

Keywords and Keyphrases

According to Feather and Sturges (1996), a keyword is any word which accurately and precisely describes the subject discussed in the document. Similarly, librarians use *subject headings* which help them identify, group and organize documents based on their similarities and make it easier to retrieve them later. A keyphrase is a multiword lexeme as against a keyword, which is a single word. Keywords and keyphrases are either selected from a standardized repository of descriptors called as a *controlled vocabulary* or are extracted from the body or more commonly, the title of the document.

One more advantage of keyphrases is their multi-functionality. According to Jones and Mahoui (2000), keyphrases and keywords can be used for natural language processing applications such as text clustering, classification, quick topic search, automatic summarization of textual matter, thesaurus construction and outputting search results. Since the efficiency of these approaches depend on keyphrases and keywords, it is important that the indexers (whether professional

humans or automatic algorithms) identify keyphrases and keywords accurately and optimally

Semantic Web

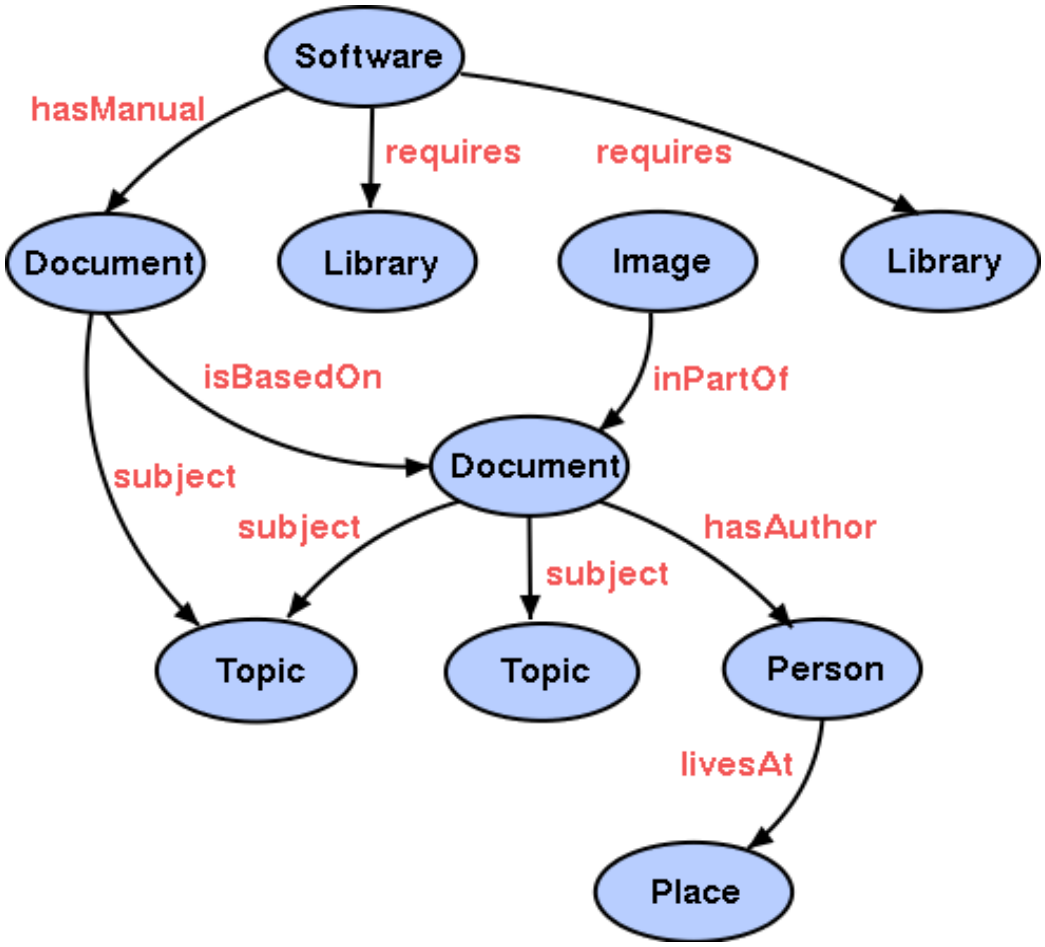

Figure 1.1 Semantic Web. Taken from W3C website http://www.w3.org/

The Semantic Web is a brainchild of Time Berners-Lee, the inventor of the World Wide Web. According to him, the Semantic Web is like a mesh of data, with all the elements interlinked to each other in such a way that the data is globally discoverable and is able to be processed by machines. The idea behind the Semantic Web is to work with the underlying semantics of the data rather than just the data itself and to achieve the same functionality that HTML did for Web 2.0, which is to

provide sufficient database. This would probably help to convert the web from a large hyperlinked data repository into a large interlinked database.

To achieve this, a major requirement is to have a consistent database so that all the data can be accessed and linked together. Resource Description Framework (RDF), Web Ontology Language (OWL) and Extensible Markup Language (XML) are the languages most widely used for the Semantic Web since these languages make it easier for machine reading and retrieval of data.

A few technologies provided by the Semantic Web are the Resource Description Framework (RDF), the Web Ontology Language (OWL), Friend Of A Friend (FOAF), Really Simple Syndication (RSS) and the Simple Knowledge Organization System (SKOS).

SKOS

The Simple Knowledge Organization System (SKOS) is a language used widely for the Semantic Web, for building controlled vocabularies which include thesauri, classification schemes such as the Dewey Decimal Classification system and subject heading systems such as LCSH (Library of Congress Science Headings). Semantic Web applications use different SKOS data which is merged and integrated with retrieval systems to increase performance in information retrieval across various collections. It is an application of the Resource Description Framework (RDF) which is a machine readable language.

SPARQL

SPARQL is a query language which allows the user to run queries and retrieve information from an RDF document. It works much the same way as any other relational query language such as SQL (Miles & Pérez-Agüera, 2007). The SPARQL protocol is a set of network protocols for interacting and retrieving data from a SPARQL binding. However, in case of SPARQL, query optimization is not supported. To make up for this, other methods of optimization can be used to provide faster access to data such as using hash tables to map string values representing concepts in the document and thus providing an entry to the vocabulary structure. These specialized functionalities are encapsulated within a custom programmatic interface, which can be bound to concrete protocols such as HTTP and SOAP.

Keyphrase Extraction

Keyphrase extraction is the process of identifying words or character strings that can be indexed as keyphrases for the document. These keyphrases should be selected from a controlled vocabulary. The keyphrase extraction method consists of two steps. In the first step, the stop words are removed from the document and the content words (non-stopwords) are identified and extracted. In the second step, these selected candidates are analyzed and the keyphrases are identified among them.

Keyphrase Extraction Algorithm (KEA)

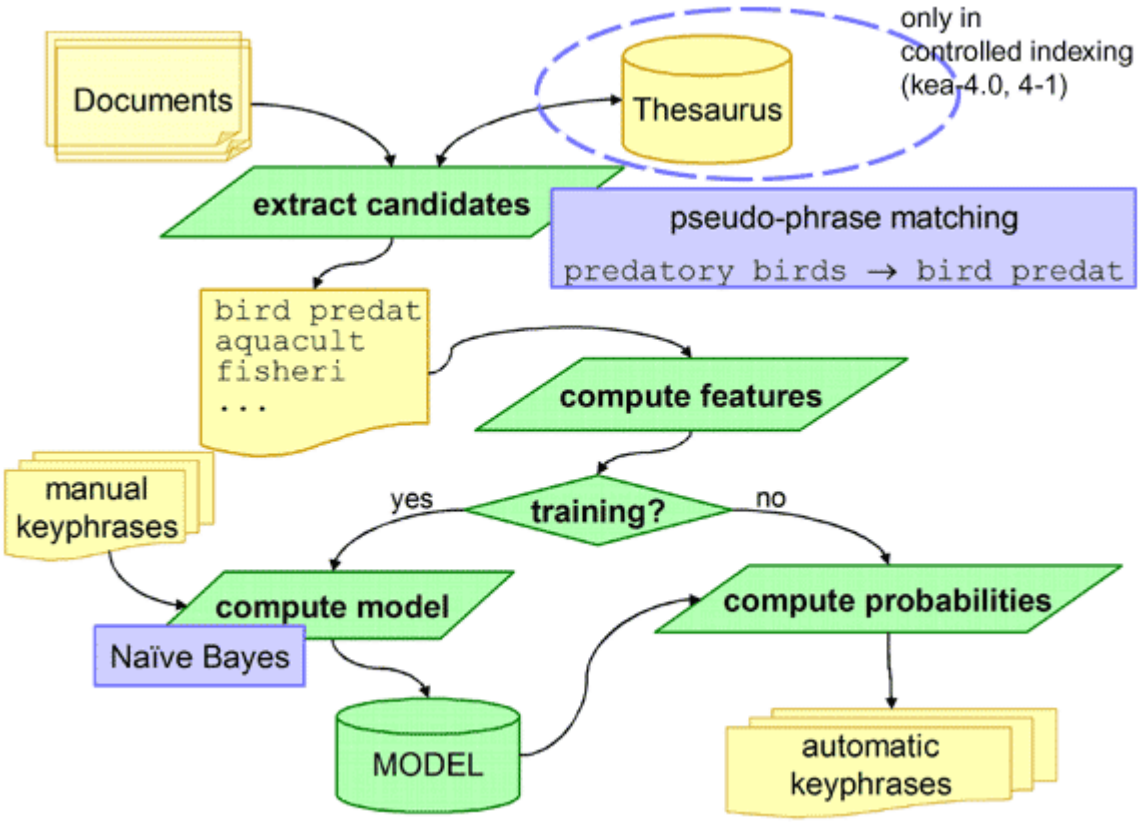

Figure 1.2 KEA Extraction Method taken from the KEA website: http://www.nzdl.org/Kea/

Keywords or keyphrases are semantic metadata elements which describe the content of the document conceptually. The Keyphrase Extraction Algorithm (KEA) is an algorithm for extracting keyphrases from text documents. It can be either used for free indexing or for indexing with a controlled vocabulary. It is open source software available to be downloaded and used under the GNU license. The working of the KEA algorithm is described as follows.

KEA uses a Machine Learning approach for automatic keyphrase extraction which consist of two phases: First is the training phase, where examples of solutions are used to explain to the system how the problem can be resolved. In our case

these solutions are documents indexed by human indexers using controlled vocabularies. From now on, this set of documents will be called the *training set*.

The second phase is the test phase, where the system tries to solve new problems which are similar to the solved problems used to train the system. In our case, these problems to solve will be the documents that we want to index using the vocabularies that we have in HIVE.

KEA uses a Machine Learning scheme that can be divided in the following parts:

1. Candidate identification using a set of features. This includes candidate extraction and feature identification
2. Filtering which consists of training the model using the training set and extracting keyphrases from new documents that are not present in the training set

1. Candidate identification

Candidate identification is the process where the most representative keyphrases are identified for each document in the training set.

1.1 Candidate extraction

Candidates are extracted from documents using some simple methods like n-gram identification, stopwords filtering and stemming. KEA calls these candidates pseudo-phrases. These pseudo-phrases are normalized using controlled vocabularies. For example, phrases such as “algorithm efficiency”, “the algorithm’s efficiency”, “an efficient algorithm” and even “these algorithms are very efficient” are matched to the same pseudo phrase “algorithm effici”, where “algorithm” and

"effici" are the stemmed versions for the corresponding full forms. The result is a set of candidate index terms for a document, and a count of their occurrences. As an optional extension, the set is enriched with all terms that are related to the candidate terms, even though they may not correspond to pseudo-phrases that appear in the document. For each candidate its one-path related terms, i.e. its hierarchical neighbors, and associatively related terms (RT), are included. If a term is related to an existing candidate, its occurrence count is increased by that candidate's count. For example, suppose a term appears in the document 10 times and is one-path related to 6 terms that appear once each in the document and to 5 that do not appear at all. Then its final frequency is 16, the frequency of the other terms that occur is 11 (since the relations are bidirectional), and the frequency of each non-occurring term is 10. This technique helps to cover the entire semantic scope of the document, and boosts the frequency of the original candidate phrases based on their relations with other candidates.

In both cases—with and without related terms—the resulting candidate descriptors are all grammatical terms that relate to the document's content, and each has an occurrence count. The next step is to identify a subset containing the most important of these candidates.

1.2 Feature identification

KEA uses the following features:

- TF×IDF

- First occurrence position of the keyphrase in the document normalized by document length
- Length of the keyphrases (in words, which boots multiterms)
- Node degree which reflects how richly the term is connected in the thesaurus graph structure.

The "degree" of a thesaurus term is the number of semantic links that connect it to other terms—for example, a term with one broader term and four related terms has degree 5. KEA considers three different variants: the number of links that connect the term to other thesaurus terms, the number of links that connect the term to other candidate phrases and the ratio of the two. Appearance is a binary attribute that reflects whether the pseudo-phrase corresponding to a term actually appears in the document.

2. Filtering

2.1 Training the model using the training set

KEA uses a classifier to create a model for 'good' and 'bad' keyphrases. In Machine Learning clustering and classification are very common techniques. In this case we have a classifier. Classifiers can be understood using the same sense that we use when we classify books. In these example we have to classify keyphrases in only two classes: *good* Keyphrases (these keyphrases that were used by human indexers in our training set) and *bad* keyphrases (these keyphrases that, although appear in the documents that we are using to train the system, have not been used by our human indexers to index the documents). These kinds of binaries classifiers are very

common in automatic classification and there exists a many methods to classify elements. KEA use a method called Naïve Bayes, which is very simple and effective. When we finish classifying the keyphrases that appear in our training set, we have a model that we can use to estimate the probability that a keyphrase is a good or a bad keyphrase for a new document, based on the model that we generated using our training set.

2.2 Extracting keyphrases from new documents that are not present in the training set

Once the model has been generated, we can offer our system to the users to index new documents. To index a new document, we only need to know the probability that have the keyphrase that occurs in this new document to be a good or a bad keyphrase. The computation of this probability is quite simple. We need to compute both probabilities based on the information which we have in our model and in the new document. So, suppose just the two features, TF×IDF and position of the first occurrence are being used. When the Naïve Bayes model is used on a candidate pseudo phrase with feature values t and f respectively, two quantities are computed:

$$P[\text{ yes }] = (Y / Y + N)\ P\_tfidf[\text{ t} \mid \text{yes }]\ \ P\_distance[\text{ f} \mid \text{yes }]$$

and

$$P[\text{ no }] = (N / N + Y)\ P\_tfidf[\text{ t} \mid \text{no }]\ \ P\_distance[\text{ f} \mid \text{no }]$$

where Y is the number of positive instances in the training files—that is, author-identified keyphrases—and N is the number of negative instances—that is, candidate phrases that are not keyphrases.

The overall probability that the candidate phrase is a keyphrase can then be calculated as: $P_k = P[\text{ yes }] / (P[\text{ yes }]+P[\text{ no }])$

Candidate phrases are ranked according to this value.

2. Literature Review

Since this study aims at studying the performance of the Automatic Indexing Algorithm, this literature review covers some literature about automatic indexing and metadata concepts, in addition to the Semantic Web.

Metadata and Digital Information

This article talks in depth about how metadata is a crucial part of machine learning and the semantic web. Greenberg defines metadata as data about data, information about information, data about information or information about data. Author, title, extension etc. is a metadata about a document. The article talks about the family of standards comprising of metadata standards and the process of metadata generation and its applications. According to Greenberg, term frequency algorithms are used to identify keywords and automatic noun detection helps identify metadata values such as author name, title etc. There are three ways of generating metadata. The first way is using derivation. Metadata is automatically derived from values associated with a document such as “date created,” “date modified” etc. The second way is by metadata extraction using the extraction process described above. The third way is called metadata harvesting which means reusing metadata which is already generated, either by automatic or manual methods.

Simple Knowledge Organization (SKOS)

This article explains SKOS, which is a formal Semantic Web language used for representing controlled structured vocabularies such as taxonomies, thesauri, classification schemes and subject heading schemes and to apply these systems for resource collection. According to Miles and Pérez-Agüera (2007), Semantic Web applications can acquire and merge SKOS data to integrate and enhance retrieval service across various collections such as libraries. The controlled structured vocabularies are intended to be used within a retrieval system and are used to "describe items in a controlled way, allowing semantically precise and unambiguous retrieval." Since it is an application of RDF, any controlled structured vocabulary represented in SKOS can be read by machines. Such vocabularies can be easily linked to other data sources. The authors talk about SKOS and then they give three examples of SKOS and RDF in terms of taxonomy, a thesaurus and a classification scheme. In the later part of the article, the authors explain how to make a controlled vocabulary available within a distributed environment using SKOS.. According to them, one simple way is to publish the entire SKOS vocabulary in a single RDF/XML document on an HTTP server so that another component within the environment can retrieve the entire document using an HTTP GET request. A more practical approach for dealing with large vocabularies is to make the SKOS representation available as SPARQL Query. SPARQL Query is an RDF query language which allows querying of data from one or more RDF graphs and works very similar to the query language, SQL.

3. Research Objectives

Metadata is one of the key elements required for retrieval of data. However, metadata search can only provide limited access, in that the user has a fixed number of options to search by, such as title, author, date of publication etc., precisely the metadata whose values have already been identified and associated with the document. On the other hand, keyphrases have an advantage over metadata in that they allow users to search conceptually. The user's choice of search is not limited by a fixed number of fields.

Sometimes, the document's authors assign the keyphrases so as to make it easier for information retrieval, especially for conference proceedings etc. In absence of any such assignment of keyphrases, professional human indexers use a controlled vocabulary to manually identify and assign keyphrases. In many cases, even if the document has been assigned keyphrases by its author, professional indexers reassign keyphrases so as to maintain consistency across documents. However, manually assigning high quality keyphrases is expensive and time consuming. For each document, the indexers have to scan it carefully and extract keyphrases which match the controlled vocabularies and are in accordance with the respective library's standards. Considering the ever growing number of digital documents on the Internet and the increasing popularity of digital libraries, having human indexers identify and assign keyphrases to documents is highly impractical.

This gives rise to the need of automatic indexing algorithms which will do the job of indexing and identifying keyphrases more efficiently but by retaining the accuracy of human indexing.

Automatic indexers available are based on simple statistical algorithms that identify keywords and keyphrases solely by comparing the frequencies of occurrence of character strings. However, very rarely do these automatic indexers make use of semantic knowledge. They blindly choose keywords and keyphrases without understanding or taking into account their actual meaning or relationships between them.

The objective of this research is to study and understand the behavior of the Keyphrase Extraction Algorithm (KEA) for automatic keyphrase extraction and how effective the indexing is in different kinds of collections. The details of the study are explained in the next chapter. The algorithm aims at achieving a higher degree of efficiency by combining the processes of extraction and assignment. The keyphrases are extracted from the document but are selected from a controlled vocabulary. Furthermore, it tries to analyze the underlying semantics of data and the relationships between them.

4. Methodology and experiments

For this study, we used the HIVE Vocabulary Server, which is a joint project of the Metadata Research Center at SILS, UNC Chapel Hill and the National Evolutionary Synthesis Center in Durham, NC.

HIVE

The Helping Interdisciplinary Vocabulary Engineering (HIVE) project aims at "dynamically integrating multiple controlled vocabularies that are encoded with the Simple Knowledge Organization System (SKOS) which is a World Wide Web Consortium (W3C) standard." The HIVE project has three components to it:

1. To provide easy and controlled yet efficient access to various controlled vocabulary for metadata generation
2. Providing newer and more efficient technical solutions to archivists, librarians and museum professionals for developing and using controlled vocabularies
3. To continuously test and improve HIVE by applying it to the Dryad repository, which is a digital repository linking data objects supporting published research.

The screenshot of the HIVE in Fig. 4.1 gives an idea of what the interface looks like. It allows the user to select one or more controlled vocabularies of the

listed LCSH (Library of Congress Subject Headings) which is a homogenous vocabulary, NBII which is a domain specific vocabulary related to the field of biology and Agrovoc which is also a domain specific vocabulary related to the agriculture field. The user can either upload a document that he wishes to generate keyphrases for, or just enter a URL of the document.

Figure 4. 1 HIVE Web Interface

Description of the study

The machine learning process was carried out by the use of two sets of documents, one containing fifty LCSH documents retrieved from Wikipedia for the training set and twenty five similar documents for the test set. The algorithm that we are using in HIVE is KEA++ which has been developed at the University of Waikato.

We randomly selected fifty Wikipedia articles for the training set and twenty five Wikipedia articles for the testing set. For each document, we selected all content specific to the article from the Wikipedia page and pasted it in a simple text file. This

way, only the text matter was saved without the images and formatting from the original article. Then professional human indexers identified the key words and keyphrases from every such article by matching them against a controlled vocabulary. For every document, we aimed at having somewhere between five to ten different keyphrases, depending on the length of the document. These keyphrases were put in a document which was a key file (having a .key extension), but the same name as the text file containing the contents of the article. Thus, we had a set of seventy five documents for the homogenous LCSH set.

The results of the KEA++ automatic indexer are compared with the indexing done by professional indexers using a controlled vocabulary.

5. Results of the experiments

LCSH documents

We randomly selected seventy five Wikipedia articles, fifty for the training set and twenty five for the testing set. The manually assigned keyphrases were compared to the keyphrases generated or extracted by the KEA algorithm. A few of these are explained below.

For the Wikipedia article about Abraham Lincoln, the keyphrases extracted by human indexers are President, United States, Illinois, lawyer, republican. As against this, the KEA++ algortithm generated the keyphrases Lincoln (Bomber), Abraham (Biblical patriarch) in the New Testament, Republicanism, Wars, Slavery (Assyro-Babylonian law), HTTP (Computer network protocol), War and civilization, Speech, Confederation (Group of poets), Parties. Here we only have one keyphrase matching, which is "Republican" ('Republicanism' as per KEA++).

As against this, for the article about Carbon, the automatically generated keyphrases are Diamonds (Shape), Graphitization, Carbonization, HTTP (Computer network protocol), Digital Object Identifiers, Retrievers, Logical atomism, Fullerenes, Isotopes, Nanotubes and the human identified keyphrases were 'nonmetallic', 'graphite', 'abundant element', 'C', 'carbon dioxide', 'atomic number 6'.

For the article on Marshalsea, the human identified keyphrases are English history, England, Great Britain, United Kingdom, UK, London, prison. The KEA++

algorithm generated the keyphrases Prisons, Boroughs (Municipal subdivision), Jails, Courts--Islamic, High Street (Perth, Scotland), Ginger, London (England)--In motion pictures, Imprisonment, Shilling, SIR (Information retrieval system). The keyphrases in this case have more similarity to their human generated counterparts.

6. Discussion of the results

When we compared the keyphrases generated as results of the automatic indexer KEA to the keyphrases generated by professional human indexers, we found that a lot of keyphrases were incorrect or not in accordance with the human generated ones. Though the algorithm worked quite well with some documents (such as 'Marshalsea'), with a few others, it did not work very well. In most cases, this was true with documents that were unrelated to the English vocabulary. In many cases

For some of the documents, it generated a keyphrase "HTTP (Computer network protocol)" which it probably generates from the word 'http' that it comes across when there are links mentioned at the bottom of the document. For example, in some documents, Wikipedia has mentioned links to YouTube videos about the document or links to some other online sources, most of which begin with "http." The KEA algorithm falsely interprets this as a potential keyphrase. This is true of the document about 'Marshalsea' as there are no http links in the document and as we expected, the algorithm did not generate a keyphrase for it. However, in the document for 'Vedas', in spite of there being http links in the Wikipedia article, no keyphrase for HTTP is generated. This, in fact, is a desirable behavior and consistency should be achieved in all the other documents where the algorithm

generates an HTTP keyphrase although none of the content matter is about the HTTP protocol in itself.

In all of the documents that we tested, the algorithm did not generate any keyphrases that were verbs or adjectives. It only generates noun phrases as keyphrases. Also, in a few documents, the algorithm generated incorrect keyphrases. For example, in the Wikipedia article for 'Bill Gates', the algorithm generated a keyphrase saying "Bill (Fictitious character: De Paola)," which has nothing to do with the document about Bill Gates in particular. In this particular case, the KEA algorithm also generated a keyphrase "IBM," but there was no keyphrase saying "Microsoft" when Bill Gates is the founder of Microsoft and is more likely to be associated with the word 'Microsoft' than 'IBM'.

In a few other cases, such as the Wikipedia article for "Vedas," the algorithm produces results which are very accurate and match to a high degree with the human generated keyphrases (seventy five percent results match the human generated keyphrases).

7. Conclusion

Automatic indexing is very much in demand, given the ever growing number of digital documents and data on the Internet and their popularity. More and more people today turn to the Internet for gathering information and hence, Internet search engines are being used more widely to retrieve this information. The automatic indexer KEA provides the user to search conceptually and thematically rather than with search strings that match the metadata elements of the document or the actual words that occur in the document. At this stage, the study shows that the automatic indexer still needs to be improved with the indexing of only the content related keyphrases.

Other Applications

The KEA indexing algorithm can be used to solve other real world problems. One of these is tagging in social networks. On social networking sites such as Facebook, Orkut, MySpace etc., social tagging is widely used. These sites allow you to tag people in pictures, videos, comments etc. It is a way of linking up that document to the tagged person's profile. By extending the functionality of automatic indexing to these features, we could allow people to search thematically rather than just the metadata associated (which in this case is just the user's profile name).

Many times, authors of papers forget to include citations or leave them out. The KEA algorithm can be also used for finding and suggesting citations and

correcting and matching the ones that are wrong. The algorithm can be used to automatically generate digital archives.

There are other features of automatic indexing that can be used for social networking such as a friend finding tool where you can search for someone's profile by extending the conceptual search functionality to search by IM names, common friends etc.

According to Medelyan (2005), KEA can also be used for semi-automatic indexing where the automatic indexer selects keyphrases and then a human indexer selects the top most appropriate ones to be associated with the document. This saves time spent in pre scanning the document on the part of the human indexer.

Limitations of KEA

The algorithm has a few limitations such as when the algorithm generates keyphrases, it considers some data which is not actually a part of the document, but is extracted from the unrelated part of the website. Also, many of the KEA generated keyphrases are not the same as the human generated ones, in that KEA misses out the important ones that are considered one of the most common keyphrases. More often, KEA generates wrong keyphrases that in fact do not tell the user what the document might be about.

According to Medelyan (2005), most of KEA's errors are related to messy inputs and erroneous stemming and can be overcome.

Limitations of the study

The study considered only homogenous documents randomly selected from Wikipedia for the testing of the KEA algorithm. This study used only the Library of Congress Subject Headings (LCSH) for generating the keyphrases. It does not consider any other domain specific documents that might generate different results. It may be that KEA works better for domain specific documents such as National Biological Information Infrastructure (NBII) for biology specific or Agrovoc for agriculture related documents. Hence, this study cannot comment on the domain specific indexing behavior of KEA.

Acknowledgments

I would like to thank my advisor, José Ramón Pérez-Agüera for his continuous encouragement and valuable guidance during this study, without which the paper would not have been possible. I am also grateful to Prof. Jane Greenberg for helping me select a field for my research and for pointing me to the right person for advising me for the completion of this paper.

References

Barbara Krasner-Khai (2001, October/November). Survivor: The History of the Library. *History Magazine*

Brin, S. & Page, L (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems.* 30(1-7), 107

Feather, J. and P. Sturges (1996). *International Encyclopedia of Information and Library Science.* London & New York: Routledge.

Greenberg, J. 2003. Metadata and the World Wide Web. *Encyclopedia of Library and Information Science*, pp. 1876-1888. New York: Marcel Dekker, Inc.

Jones, S. and M. Mahoui (2000). Hierarchical document clustering using automatically extracted keyphrases. In *Proc. of the 3rd International Asian Conference on Digital Libraries*, pp. 113–120.

Miles & Pérez-Agüera, (2007). Simple Knowledge Organisation for the Web. In J. Greenberg & E. Mendez (Ed.) *Knitting the Semantic Web*. pp 69-83. Haworth Press, Incorporated.

Olena Medelyan (2004-2005), Automatic keyphrase indexing with a domain-specific thesaurus

Silverstein, C., M. Henzinger, H. Marais, and M. Moricz (1998). Analysis of a very large AltaVista query log. Technical Report 1198-014, Digital SRC.

Websites

Franklin, C. *How Internet Search Engines Work.* Retrieved from How Stuff Works? Website: http://www.howstuffworks.com/search-engine.htm

Library. (n.d.). In *Wikipedia*. Retrieved March 5, 2010, from website: http://en.wikipedia.org/wiki/Library