

Synonymous Searching in an Accessible Environment Using Proximal Keywords

Jason Hines
Mathematics and Computing Science
Saint Mary's University
Halifax, Nova Scotia, B3H 3C3
Canada

j_hines@cs.smu.ca

ABSTRACT

Accessibility in the classroom is an ever-growing problem due to increasingly diverse student populations, hyper-growth in technologies such as the Web, and newer more complex document representations. Using speech recognition software, the Liberated Learning Consortium attempts to solve this problem by providing students with multimedia enhanced lecture transcripts. Unfortunately, retrieval of such a document can be a complex process due to many interrelated factors not limited to, but including, learning impairments. This paper describes a Web-based system for effectively retrieving keywords, keyphrases, and documents. The system uses synonym and proximity lists of keywords as an alternative to traditional keyphrase representations.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *Retrieval models, search process, selection process.*

General Terms

Management, Design, Experimentation, Human Factors, Languages.

Keywords

Accessibility, Liberated Learning, Speech Recognition, Search and Retrieval.

1. INTRODUCTION

The Liberated Learning Consortium (LLC) is an international effort to advance speech recognition technology and techniques to create and foster barrier-free learning environments where all learners have equal access to information. LLC approaches this problem by using specially designed speech recognition technology to create automatically transcribed text and multimedia documents.

Copyright is held by the author/owner(s).
ASSETS'06, October 22–25, 2006, Portland, Oregon, USA.
ACM 1-59593-290-9/06/0010.

Effective retrieval of these documents can play an important role in the success of the LLC, however, common course management systems like WebCT and Moodle do not provide the flexibility needed to effectively search through such documents and therefore a LLC specific system is required.

Information retrieval researchers have developed automatic indexing techniques to extract keywords from documents. However, in some environments, keyphrases may be a more attractive alternative. There are many keyphrase generation algorithms that replace keywords with keyphrases but this paper discusses a more flexible approach that generates keyphrases based on the proximity of keywords. To aid in the automatic generation of keyphrases, the system uses the Paice/Husk stemming algorithm which converts all the variant forms of a word to the same standard form in most cases [1]. Using the proximity information, the system is able to dynamically produce a potential list of keyphrases and keywords with similar stems. The proposed approach is useful for spoken language documents where the order of the words in a keyphrase is not always consistent e.g. “proof by mathematical induction”, “proof by induction”, and “inductive proof”.

The paper discusses one existing work on keyword and keyphrase extraction (section 2) and provides a short introduction to the proposed representation and generation methods (section 3) followed by a conclusion (section 4).

2. EXISTING KEYPHRASE RESEARCH

There are two fundamentally different approaches to keyphrase generation: keyphrase assignment and keyphrase extraction [3]. Witten, et al.[3] proposed the Kea keyphrase extraction algorithm which generates keyphrases by searching through a document for any sequence of one, two, or three consecutive words. The consecutive words cannot be separated by any punctuation and cannot begin or end with stop words. Stop words are commonly occurring words that have little semantic value such as “to”, “the”, “of” and “he”. Candidate phrases are normalized by converting them to lowercase and stemming them. Stemming is a process of linguistic normalization where variant forms of a word are converted to a standard form, for example, the words “access”, “accessible”, and “accessibility” all stem to “access”. Since the process is aimed at mapping for information retrieval purposes, the stem need not be a linguistically correct lemma or root [1]. Kea then uses an algorithm to classify the candidate phrases as keyphrase or nonkeyphrase. A study has shown that Kea extracts good keyphrases, as measured by human subjects and their assessments were uniformly positive [2].

3. SEARCHING AND INDEXING FOR LLC

The proposed system provides searching abilities based on keyphrases and stemming, automatic phrase table generation, listing of all the course documents, and the ability to crawl and index external websites containing various file types. The system updates automatically on a scheduled basis.

3.1 Phrase Table Generation

The proposed system differs from Kea in a number of ways. A phrase in our system is always two words. Each indexed word has two lists associated with it; a synonym list (Figure 1), consisting of words which have the same stem, and a proximity list (Figure 2), consisting of words which appear within a threshold distance of +/- five words. A distance of five words was based on the assumption that a phrase will not be more than five words long, whereas Kea phrase extraction only considers phrases up to three words long which do not begin or end with a stop word. It should be noted that the concept of synonyms within this paper is different than the actual definition; here it merely represents words that have the same stem. The synonym list is useful for displaying the various forms of a word in the phrase table. The proximity list consists of quadruplets (keyword, frequency, average distance, minimum distance) which are used to determine the relevance of keyphrases. Using these two lists, a software agent automatically generates a phrase table each time the system is updated. The phrase table (Figure 3) gives users the ability to see the important phrases which exist within the document set and also provides hyperlinks to quickly search for those phrases.

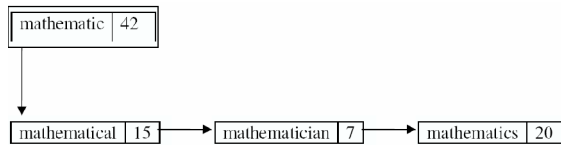


Figure 1. Synonym List

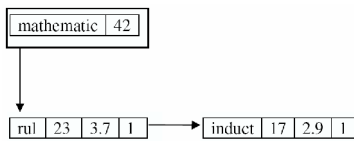


Figure 2. Proximity List

address	Similarly Spelled	Note Finder	Google Search
issues	Similarly Spelled	Note Finder	Google Search
africa	Similarly Spelled	Note Finder	Google Search
asia	Similarly Spelled	Note Finder	Google Search
south	Similarly Spelled	Note Finder	Google Search
again	Similarly Spelled	Note Finder	Google Search
ago	Similarly Spelled	Note Finder	Google Search

Figure 3. Phrase Table

3.2 Calculating Relevance

The relevance calculation is an extension of the vector space model as opposed to the Naïve Bayes model used in Kea. Equation 1 shows a formula for calculating the relevance of a document represented using keyphrases.

$$\begin{aligned}
 \text{relevance} = & w_s \times \sum_{q_i} \text{freq}(q_i, d) \\
 & + w_p \times \sum_{q_i, q_j, i < j} \text{freq}(q_i + q_j, d) \\
 & + w_a \times \sum_{q_i, q_j, i < j} \text{avgDist}(q_i, q_j, d) \\
 & + w_m \times \sum_{q_i, q_j, i < j} \text{minDist}(q_i, q_j, d),
 \end{aligned}$$

Equation 1.

Where q_i and q_j are keywords in the query, d is the document, $\text{freq}(q_i, d)$ is the frequency of q_i in document d , $\text{freq}(q_i + q_j, d)$ is the frequency of q_j in the proximity list of q_i , $\text{avgDist}(q_i, q_j, d)$ is the average distance between q_i and q_j , while $\text{minDist}(q_i, q_j, d)$ is the minimum distance between q_i and q_j . The weights w_s , w_p , w_a , and w_m signify relative importance of the four equations in Eq. 1.

Although this calculation provides additional storage and computing overhead, it will improve the quality of ranking. The loss of storage and computational efficiency will be relatively small when applied to a single course, which usually has less than one thousand documents.

4. CONCLUSION

The proposed system was tested in a classroom environment and received promising feedback of which over eighty percent was positive. Many improvements have since been made.

Stemming in the English language has been a long fought battle and it is not completely accurate. In some instances many keywords which are obviously non-related will have the same stem. However, some of this effect can be masked by introducing new stop words and by using relatively small document sets.

For a complete online demo of the proposed system please see <http://notefinder.smu.ca/libdemo>.

5. ACKNOWLEDGMENTS

Special thanks to the Liberated Learning Consortium and the Natural Sciences and Engineering Research Council of Canada for providing funding for this research, and Dr. Pawan Lingras for providing support.

6. REFERENCES

- [1] Paice, Chris D., *Another stemmer*, ACM SIGIR Forum, v.24 n.3, p.56-61, Fall 1990
- [2] Steve Jones, Gordon W. Paynter, *Human evaluation of Kea, an automatic keyphrasing system*, Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries, p.148-156, January 2001, Roanoke, Virginia, United States
- [3] 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, New Zealand, 2000