World Applied Sciences Journal 17 (5): 611-616, 2012
ISSN 1818-4952
© IDOSI Publications, 2012

Efficient Cluster-Based Information Retrieval from Mathematical Markup Documents

[1]*Muhammad Adeel,* [1]*Muhammad Sher and* [3]*Malik Sikandar Hayat Khiyal*

[1]Department of Computer Science, Faculty of Basic and Applied Sciences,
International Islamic University, Islamabad, Pakistan
[2]Department of Computer Science and Software Engineering,
Faculty of Science and Technology, Fatimah Jinnah Women University, Rawalpindi, Pakistan
[3]APCOMS, Khadim Hussain Road, Lalkurti, Rawalpindi, Pakistan

Abstract: The paper presents the innovative use of cluster based search for efficient mathematical information retreival. The search is realized by applying multiple clustering techniques on the mathematical markup documents. The technique makes use of cluster oriented search to speed up math information retrieval. Impressive results have been obtained as compared to similarity based search. With the use of cluster based search, the retrieval time has been reduced from multiple seconds to about 1 second. The quality of the result set has been found to be comparable to similarity based search.

Key words: Cluster Based Search · Math Information Retrieval · Clustering Techniques · K-Means · Agglomerative Hierarchical Clustering · Kohonen Self Organzing Maps

INTRODUCTION

Several research efforts have been conducted recently to develop math search systems This includes work by [1-6]. Miller and Youssef [1] have developed the search system for the Digital Library of Mathematical Functions. The library extends mathematical support to existing text search techniques for math information retrieval. Kohlhase and Sucan [2], and Kohlhase *et al.* [5] have developed MathWebSearch, a mathematical semantic search engine. The system enables semantic search of mathematical content using substitution tree based indexing. Munavalli and Miner [3] have presented MathFind, an evolutionary math search engine. The technique used by MathDex is to linearize mathematical notation as a sequence of text tokens [6]. MathGO-I by [4] is a search tool for providing formula based search. MathGO-I translates math expressions into text and combines it with other text terms. The system enables search for math formula expressions and equations. With the existing work on math search systems, speed is an important issue to resolve. Work presented by Kohlhase and Sucan [2], and Kohlhase *et al*. [5] has been limited by the use of trees for indexing and retrieval purpose. The vector space model has been used by Miller and Youssef [1], and Munavalli and Miner [3] which can get slow with increasing number of documents. Math search systems need to implement some technique to speed up retrieval. The technique should speed up retrieval without compromising on quality.

Cluster based search can be utilized as a tool to improve retrieval performance for information retrieval tasks. In this technique, documents are grouped into similar clusters based on some similarity metric. A centroid is calculated as representative for each cluster. The user specified information need is compared with the centroids of clusters to identify the closest clusters. Documents belonging to the closest matching clusters are then combined, sorted and retrieved. The cluster based approach is likely to improve the search time considerably.

Related work on cluster based retrieval includes [7-10]. Cutting *et al.* [7] and Hearst and Pedersen [8] have utilized document clustering for organization and display of retrieval results in the Scatter/ Gather system. The experiments confirm the efficiency and effectiveness of cluster based retrieval. Both Can *et al.* [9], and Liu and Croft [10] have reported encouraging results with cluster based search on text data set. Liu and Croft [10] show the cluster based retrieval to perform consistently better over document retrieval. We now investige multiple clustering techniques to improve mathematical retrieval.

Corresponding Author: M. Adeel, Department of Computer Science, Faculty of Basic and Applied Sciences, International Islamic University, Islamabad, Pakistan. Tel: +92-321-5009423.

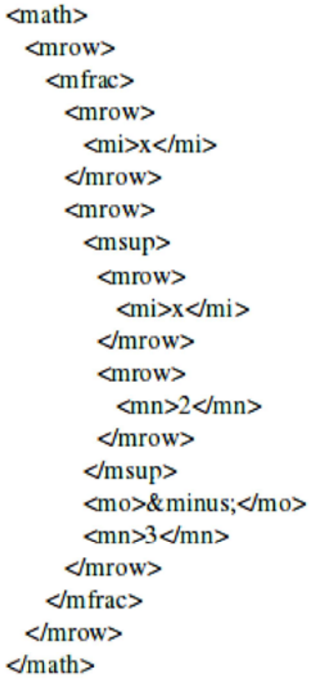
MATERIALS AND METHODS

System Description: The utilized system for our experiments is a math-aware search system by [11]. The system is able to process the data set into document vectors using the vector space model. The documents in the data set may consist of text as well as math markup. The standard for math markup is the mathml[1]. By the use of this standard, universal mathematical document processing and exchange is possible. The text part of the document is translated into token after preprocessing steps. The math part is translated into tokens using regular expression matching and keyword extraction. All identified tokens are assigned count based on their occurence. The tokens are put into a vector which becomes the representative of the original document. Once all documents are converted into vectors, weights are assigned with vector space model. The document vectors with weights assigned are now prepared for retrieval purpose.

Math formula/expression is typed into the system using the graphical formula editor WebEQ (WebEQ editor, http://www.dessci.com/)[5]. A query may consist of text as well as math mark up tags. A given mathematical query is processed into a vector with the same process above. The query vector is like any other document vector for comparison purposes. The math markup of a sample mathematical document "Differentiate, with respect to x, x/(x^2-3)" is indicated in Figure 1.

The query vector is now compared with all document vectors using euclidean distance metric. The closest vectors with the query vector are identified and their corresponding documents are retrieved. The paper presents the use of cluster based search with the above system to speedup math information retrieval. The clustering algorithms tested include k-means, agglomerative hierarchical clustering and Kohonen self organizing maps.

Document Clustering: Given the complex nature of mathematical documents, ordinary similarity search can be prohibitively slow. In this section, three clustering techniques are presented which have been investigated to improve the retrieval experience i.e., K-Means clustering, Kohonen Self Organizing Maps (KSOM) and Agglomorative hierarchical clustering (AHC).



```
<math>
  <mrow>
    <mfrac>
      <mrow>
        <mi>x</mi>
      </mrow>
      <mrow>
        <msup>
          <mrow>
            <mi>x</mi>
          </mrow>
          <mrow>
            <mn>2</mn>
          </mrow>
        </msup>
        <mo>&minus;</mo>
        <mn>3</mn>
      </mrow>
    </mfrac>
  </mrow>
</math>
```

Fig. 1: MathMarkup of a Selected Mathematical Document

The documents are clustered by using the user selected algorithm and the similarity of the query vector with the centroids of all generated clusters is computed. The outcome is that the most similar clusters are selected. The question vector entries (data points) are sorted. The process is outlined in detail in Table 1. The results from the closest clusters are displayed in decreasing order of proximity.

K-Means Algorithm: The K-Means algorithm first introduced by Hearst and Pedersen [8], is used to classify a given data set into predefined clusters. The algorithm belongs to the class of divise partitioning algorithms. The algorithm starts by defining k centers in the data. All points are then assigned to their nearest centroid. This initial step is followed by centroid reestimation and again point allocation. The procedure continues until centroid do not make any more significant change [9]. The algorithm is terminated after predefined iterations or through the objective function [9]. Given its linear complexity O(n), the algorithm is computationally feasible. The k-means aims to minimize the objective function which is the mean squared error. The objective function [9] is

[1] Mathml 2.0, a w3c recommendation. http://www.w3.org/tr/mathml2/, Oct. 2003

Table 1: Cluster Enabled Search

```
Algorithm:    Cluster Enabled Search GetClosestCentroids
Input:
QueryVecMin - Normalized Question Query Vector
DVEC = {di - 1 ≤ i ≤ n } - set of all math equation vectors stored in the database clusterCentroids - two dimensional array to hold cluster centroids and associated values cNodes - one dimensional array to hold cluster centroids and associated values matchCount - number of results requested by the user
clusteringAlgo - clustering algorithm previously selected by the user
Output:
RES - array consisting of question ID's that would contain ?nal set of results requested
by the user.
Process:
Initialize the cNodes array (Create objects for all centroid positions).
forall c_i□ cNodes do
    Create a new object for this index of cNodes
    Record position of this index in cNodes
    Store ClusterID of this centroid node in cNodes array
end for
if cluster has some data points do
    Calculate distance of the centroid with the query vector
end if
for i = 1 to n do
    sort cNodes array in ascending order of values
end for
for i = 1 to 10 do
    Obtain all the question IDs of the current cluster
    Calculate the distance of all questions with this cluster with the query vector
    Sort data points in this cluster according to distance with the query vector
    if (currentresultsize ¡ matchcount ) Copy the sorted data points into RES theresultsarray
end for
{RES now contains the required vectors}
```

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_i^{(j)} - c_j \| \qquad (1)$$

Where $x_i^{(j)} - c_j$ is a chosen distance measure between the point $x_i^{(j)}$ and the cluster center c j.

The algorithm is composed of the following steps [9]:

- Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
- Assign each object to the group that has the closest centroid.
- When all objects have been assigned, recalculate the positions of the K centroids.

Agglomerative Hierarchical Clustering: Agglomerative hierarchical clustering is a case of hierarchical clustering techniques. The technique works as repeatedly clustering the documents from top or bottom. We have investigated this technique for math information retrieval. The tree formed by this technique can be investigated at various levels. The technique work as follows[2].

- Start by assigning each item to a cluster, so that if we have N items, we have N clusters. Each cluster initially contains just one item.
- Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
- Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now we have one cluster less.
- Compute distances (similarities) between the new cluster and each of the old clusters.
- Repeat steps 3 and 4 until all items are clustered upto a specified threshold.

[2] Hierarchical Clustering Algorithms, http:// home.dei.polimi.it/ matteucc/ Clustering/ tutorial_html/ hierarchical.html

Hierarchical scheme has a complexity of O(n^2). This indicates a substantial time taken during the data set clustering. Due to the working nature of the algorithm, it cannot undo what was previously done.

Kohonen Self Organizing Maps: Kohonen self organizing maps are a type of artifical neural networks. Unlike other neural networks, the self organizing map is able to classify data without supervision. The map perform dimensionality reduction and display similarities between data items. Usually a one or two dimensional self organizing map is generated to understand the original data. Working of the self organizing map is as follows[3]:

Each data from data set recognizes themselves by competeting for representation. SOM mapping steps starts from initializing the weight vectors. From there a sample vector is selected randomly and the map of weight vectors is searched to find which weight best represents that sample. Each weight vector has neighboring weights that are close to it. The weight that is chosen is rewarded by being able to become more like that randomly selected sample vector. The neighbors of that weight are also rewarded by being able to become more like the chosen sample vector. From this step the number of neighbors and how much each weight can learn decreases over time. This whole process is repeated a large number of times, usually more than 1000 times.

The algorithm is composed of the following steps:

- Each node's weights are initialized. Number of nodes depend on the size of the map.
- A vector is chosen at random from the set of training data.
- Every node is examined to calculate which one's weights are most like the input vector. The winning node is commonly known as the Best Matching Unit (BMU).
- Then the neighbourhood of the BMU is calculated. The amount of neighbors decreases over time.
- The winning weight is rewarded with becoming more like the sample vector.

The nighbors also become more like the sample vector. The closer a node is to the BMU, the more its weights get altered and the farther away the neighbor is from the BMU, the less it learns.

- Repeat step 2 for N iterations.

RESULTS AND DISCUSSION

The document collection was clustered using the three clustering algorithms indicated earlier. The collection contained documents representing different math categories. The categories include topics such as Matrices, Functions and Graphs, Complex Numbers. The document collection comprising 413 documents was stored in MySQL database. For each mathematical category, representative queries have been formulated. The clustering parameters have been empirically selected to generate optimal clustering outcome. Precision has been estimated at best threshold value. The results are compared with those of similarity search which forms the baseline. The formula for calculating precision scores is indicated below:

Precision = Score of Matching Documents / Total Score of Retrieved Documents .... (1)
(A complete match gets a full score of 1 while a partial match gets a score of 0.5.).

Table 2: Precision Scores

	Similarity Based Search	K-Means	Self Organizing Map	Average-link
Differentiation	94.16%	95.83%	93.33%	77.50%
Integration	82.22%	77.77%	77.77%	81.11%
Complex Numbers	70.00%	85.00%	85.00%	85.00%
Functions and Graphs	85.00%	90.00%	90.00%	85.00%
Matrices	50.00%	10.00%	30.00%	40.00%
Sequences and Series	90.00%	80.00%	90.00%	60.00%
Trigonometry	100.00%	100.00%	95.00%	100.00%
Vectors and Geometry	100.00%	90.00%	90.00%	90.00%
Conic Sections and Polar Coordinates	60.00%	100.00%	50.00%	70.00%

[3] Self-Organizing Maps, http:// www.sis.pitt.edu/ ~ssyn/ som/ som.html

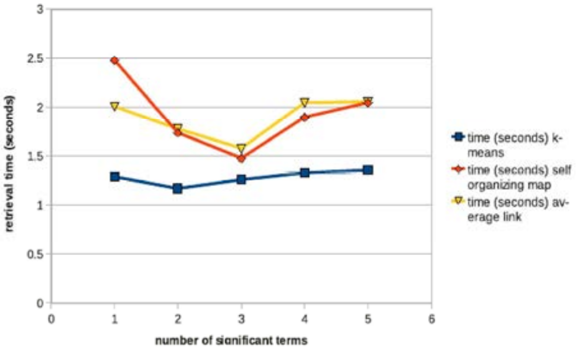


Fig. 2: Retrieval Time taken by different clustering algorithms

Table 2 presents the system retrieval performance. Four approches similarity search, kmeans based search, self organizing map based search and average link based search have been tested. For each mathematical category, representative queries have been used. The similarity search results form the baseline for other techniques. In the differentiation category, k-means is the most efficient with about 96% precision. It has even surpassed the baseline score of 94%. This behavior is likely due to formation of compact and well separated clusters in this category. In the Integration category, similarity search and average link based search are almost equally effective. Both score respectively with 82% and 81% respectively. K-Means and self organizing map scored equal scores of about 78%. In the complex numbers category, all three clustering approaches managed to outperform the similarity search approach. In functions and graphs category, k-means and self organizing map achieved highest precision scores of 90%. Similarity search and average-link managed to score 85% each. Results in the matrices category were below average for all four approaches. Investigation of questions revealed the need of adding more suitable keywords to matrices questions. Similarity search and self organizing map managed to achieve highest scores of 90% for the sequences and series section. This was followed by k-means with 80% and average-link with 60%. All techniques scored 100% for trigonometry except for 95% for self organizing map. In vectors and geometry, similarity search achieved 100% with the other approaches scoring 90% each. Finally, k-means scored 100% results on conic sections & polar coordinates.

During the training phase of data, K-Means was the fastest clustering algorithm. Average link was second best in terms of time taken for clustering data set. Self organizing map is the slowest technique among all three approaches. In a full scale system, hierarchichal clustering and self organizing map can be expensive in terms of training time. K-means deliver comparable retrieval performance to the other two approaches. Figure 2 indicates that k-means is the fastest in the retrieval phase too. The results have been calculated on Intel Core 2 Duo processor with two 1.66 Ghz cores, 1 Gigabyte (double data rate 2, DDR2) memory. The operating system utilized was Linux and the development environment was Java with Netbeans Integrated Development Environment.

ACKNOWLEDGEMENTS

The primary author gratefully acknowledges the help and support by Professor Hui Siu Cheung of Nanyang Technological University. The author acknowledges the gracious support by my supervisor, Professor Sikandar Hayat Khiyal. Moreover, the author acknowledge sponsorship and support provided by Higher Education Commission, Pakistan for this research. Finally, support extended by the author parent university is very gratefully acknowledged.

REFERENCES

1. Miller, B.R. and A. Youssef, 2003. Technical aspects of the digital library of mathematical functions. Annals of Mathematics and Artificial Intelligence, 38: 121-136.
2. Kohlhase, M. and I.A. Sucan, 2006. A search engine for mathematical formulae. In Proc. of Artificial Intelligence and Symbolic Computation, number 4120 in LNAI, Springer, pp: 241-253.
3. Munavalli, R. and R. Miner, 2006. Mathfind: a math-aware search engine. In SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA. ACM, pp: 735-735.
4. Muhammad Adeel, H., S. Cheung and M.S.H. Khiyal, 2008. Math GO! Prototype of a Content Based Mathematical Formula Search Engine. Journal of Theoretical and Applied Information Technology, 4(10): 1002-1012.
5. Kohlhase, M., S. Anca, Constantin Jucovschi, Alberto Gonz'Lalez Palomo and Ioan A. S'Cucan, 2008. Math Web Search 0.4, a semantic search engine for mathematics. manuscript, see http://mathweb.org/projects/mws/pubs/mkm08.pdf.

6. Miner, R. and R. Munavalli, 2007. An approach to mathematical search through query formulation and data normalization. In Calculemus/MKM, pp: 342-355.
7. Cutting, D., D. Karger, J. Pedersen and J. Tukey, 1992. Scatter/gather: A cluster-based approach to browsing large document collections, in Proceedings of the 15th annual international ACM SIGIR conference on Research and Development in Information Retrieval, pp: 318-329.
8. Hearst, M. and J. Pedersen, 1996. Reexamining the cluster hypothesis: scatter/gather on retrieval results, in Proceedings of the 19th annual international ACM SIGIR conference on Research and Development in Information Retrieval, pp: 76-84.
9. Can, F., Ismail Seng_Nor Alting_Novde and E. Demir, 2004. Efficiency and effectiveness of query processing in cluster-based retrieval. Inf. Syst., 29(8): 697-717.
10. Liu, X. and B. Croft, 2004. Cluster-based retrieval using language models. In Proceedings of SIGIR, ACM Press, pp: 186-193.
11. Search system for Mathematical Information Retrieval. Adapted from Professor Hui Siu Cheung, School of Computer Engineering, Nanyang Technological University, Singapore.