ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Effectiveness of Ant Colony Optimization with Hybrid distance for document Retrieval

Subhadra Kompella
Assistant Professor, Dept. of CSE
GITAM University
Visakhapatnam, India

Shashi M
Professor, Dept. of CSE,
Andhra University
Visakhapatnam, India

Abstract: In the context of document ranking the user's information requirement is often expressed partially as a query consisting of a set of keywords. The existing distance estimates though proven good for comparing two documents expressed as n-tuples, are not effective while comparing a query expressed as k-tuple (k<<n) with a document expressed as n-tuple. This proposed method deals with these two challenges by using parametric modelling for identifying plausibly relevant documents based on the query expressing the information requirement partially. This task relevant subset of documents is intensely explored to cluster them into semantically related groups. The author proposed to estimate the similarity between a pair of documents by integrating the results of micro as well as macro level analysis. A new Hybrid Distance metric is devised as a combination of frequency based distance (Macro level) and Structure based distance (Micro level). The hybrid distance is used for clustering the documents employing a scalable variant of Ant Colony Optimization that simulates the brood sorting behavior of ants with two modifications. The author proposed to use winged ants that hop instead of walk on the 2D grid where each grid cell accommodates multiple documents instead of single document to form better clusters in less number of iterations.Each Document cluster represents semantically related group of documents on a theme(s) and hence contain relevant documents for a particular information requirement based on its resemblance to the query. Documents of most resembling cluster are ranked based on their resemblance to form document ranking in response to the user's query. The performance of clustering and document ranking has been tested on standard data repository which is a subset of Wikipedia using F-measure and Mean Average Precision respectively.

Keywords: clustering; precision; recall; average precision-measure; query; relevance

I. INTRODUCTION

The enormously large increase in the usage of textual data through social media, information centric applications like scientific journals, World Wide Web and other information retrieval applications creates the need for mining relevant textual data from large data corpora. Due to the increasing usage of the data from the large databases a much of research has been carried out in the field of text mining. Text mining is a process of analyzing and retrieving text from a large text data collection. The aim of text mining is to focus on valuable information from the inner details of the documents which correlate to that of user's interest. Text mining involves different functions such as text categorization, textual clustering, theme or keyword extraction, sentiment analysis, text retrieval etc. The preliminary step of text mining as a process of information retrieval is information extraction.Text mining is a sub-field of data mining that focuses on extracting patterns from text corpora generally available as a set of documents. Information extraction is a basic task of information retrieval. An information retrieval system is a process of obtaining relevant information from the data corpora, which is extraction of relevant document from the data corpora. Information Retrieval techniques primarily use "bag-of-words" model for the tasks such as document matching, clustering, and ranking[10]. The concept of information retrieval is used in the current research areas such as artificial intelligence, machine learning, natural language processing and widely used in text clustering.

Text clustering aims at grouping similar documents into a cluster. For effective clustering to be performed an appropriate distance metric is essential to find out the proximity between a pair of documents. Euclidean distance is the common measure to find proximity in documents. In this work a variant of Euclidean distance is used to compare every pair of documents in the corpora. Euclidean distance is widely used as it obeys the symmetric property, positivity, triangle inequality and positive definiteness when applied between two document vectors and results in a value of zero if the documents are highly similar. This work has used Euclidean distance in two instances: during Micro level proximity analysis and Macro level analysis between documents from the corpora. Micro level analysis is performed using the estimation of Structure-based distance and Macro level analysis is obtained through estimating Frequency-based distance among documents referred to as *Hybrid Distance*.For any document clustering algorithm the fundamental step is to calculate the distance between the documents represented as document term vectors. Conventional distance estimates applied on a pair of document term vectors rely solely on the occurrence of a term in the whole document ignoring the location of occurrence. However pair of documents are more similar if they possess the same structure while containing same term frequencies. Hence distance estimate should distinguish documents in terms of content as well as the structure of the document. Hence in this research excercise, the author seeks to analyse the documents at macro as well as micro level: A macro level analysis considers a document as a single document term vector while the micro level analysis divides the document into multiple segments and represents each segment as a separate term vector namely document segment term vector. Macro level analysis estimates frequency based distance while micro level analysis results in structure based distance between a pair of documents. This work proposes a hybrid distance metric [12] which is a combination of frequency based measure as well as the structure based measure of the distance between a pair of documents. The following section outlines the concept of the frequency based

values are used as the inputs for document clustering.

User's query in general contains limited number of keywords/search terms and hence can only specify the user's information requirement partially. Author suggests to use this partial information provided as a small list of query terms to identify task relevant data using Poisson score parameters defining the distribution of relevant documents for the query. The documents that are plausibly relevant to a query form the task relevant data which will be further explored for ranking the documents based on their relevance to the user's query. Since the information requirement of the user is not fully specified in the query, the author suggests making use of inherent information /patterns hidden in the task relevant data/documents to identify the relevant documents for the user's information need. Partitional clustering of task relevant data extracts the hidden patterns and groups the documents that share the same pattern into a cluster[2]. The Swarm Intelligence algorithm namely Ant Colony Optimization is found to be effective for this purpose[3].The process of clustering in this paper implements a standard swarm intelligence algorithm for clustering, Ant Colony Optimization algorithm for clustering documents followed by pruning the clusters by identifying medoids of each cluster and redistributing the elements of the cluster with the closest cluster medoid.

The framework of clustering proposed in this work is explained using a flow chart given below. As is evident from the flowchart the clustering module makes use of the document vectors of task relevant documents represented on Vector Space Model and the hybrid distance matrix to make the process of clustering effective.

The entire process of this chapter is detailed diagrammatically as follows:

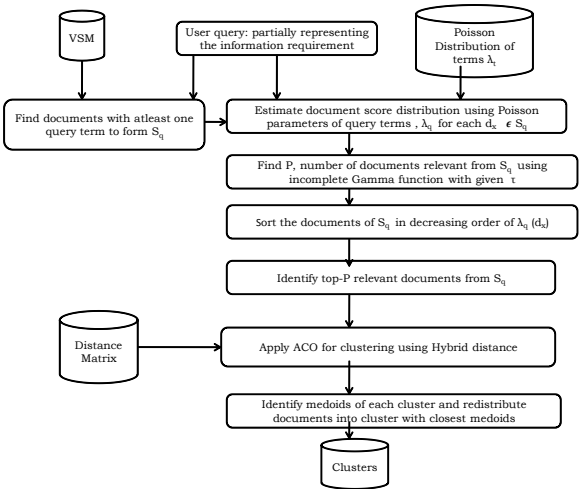

Figure 1. Screening and Clustering using ACO with Hybrid Distance Metric.

The above flow chart is described with the details of ACO clustering in the following section, which is used for further retrieval of documents through the obtained clusters.

A. Task Relevant Data Identification

- **User Query**

The query is given by the user which is a collection of terms that partially reflect the information required by the user. Depending on the resemblance of the query terms to document vectors in Vector Space Model, the resembling documents are retrieved to form a set, Sq which is a subset of the document collection. Hence Sq is a collection of documents containing query terms and which is sorted in the decreasing order of their resemblance to the user query. After discovering the resembling document set, Score Distribution of the documents for the set is estimated with reference to the Poisson parameter estimation of the Query terms, λq, which is explained in the next section.

- **Poisson Score Parameter for Query**

Score Distribution of documents is estimated in order to concentrate only on those documents that are assumed to reflect the user's information need. Considering the values of λt from the Poisson Probabilistic model proposed in [11,14], the Poisson parameter values for the query terms λq, are calculated to retrieve the most similar documents for the given query. The Poisson score distribution of the documents with respect to the user query is calculated using the formula given below:

$$\lambda_q^s = \sum_{t \in q} \left(\frac{\lambda_t}{1 - \exp(\lambda_t)} . idf(t) \right) + \sum_{t \in q} (\lambda_t . idf(t)) \quad (3)$$

From the above formula, the list of documents that match the query terms are retrieved to compose task relevant data for the query. The subset Sq is formed with a collection of documents having a score distribution higher than the user defined thresholdτ.The current framework uses the threshold

value τ =0.5. The number of documents that are relevant is calculated as described below:

$$|\text{Docs(score}>\tau)| = |D|.\left(1 - \frac{\Gamma[\tau+1], \lambda_q^s}{[\tau]!}\right) = P = P \qquad (4)$$

where Γ is a popular Incomplete Gamma function defined to obtain number of relevant documents in Sq.

An Incomplete Gamma function is a statistical measure which is used to identify a set of plausibly relevant documents based on the user defined threshold. This is defined as follows:

$$\Gamma\left([\tau+1], \lambda_q^s\right) = \int_0^{\lambda_q^s} t^{\tau}\, e^{-t}\, dt \qquad (5)$$

After the scores of all the documents with respect to the presence of the query terms are calculated, those documents that are having score greater than the user defined threshold τ are identified to form a subset of the corpora to limit the search space to P, number of plausibly relevant documents.

Identifying top-P relevant Documents

Applying Incomplete Gamma Function on the document subset which is considered as a set of plausibly relevant documents to the given user query, the contents of the subset are processed to identify the top-P relevant documents that are very much closer to the query terms. The identified plausibly relevant documents are referred to as task relevant documents and are then clustered.

Now, the task relevant documents are clustered usingAnt Colony Optimization algorithm for exploiting the inherent patterns shared by groups of documents for effective relevance analysis[5].

B. Clustering Using Ant Colony Optimization (ACO) algorithm

The proposed work implements a variant of the concept of Scatter-Gather approach [4,6] where the documents are randomly scattered on a 2D grid and then are picked up by an ant and dropped at its neighbouring document which is more similar to it based on the hybrid distance [11]. The picking of a document is done randomly from the scattered documents on the grid and the dropping is based on the hybrid distance metric. The probabilities of picking/dropping a document are discussed in detail in the next section. The main drawback of applying ACO for document clustering[7,8] is that unless it is executed repeatedly through many iterations, some of the documents may be left in their original positions as they were not picked up by any ant and hence become members of wrong clusters. In the proposed work a 'post pruning step' is applied on the results produced by ACO with limited iterations for achieving better quality clusters with minimal overhead.

Document space being high dimensional requires a better distance metric than the conventional Euclidean distance. The proposed method implements ACO with Hybrid distance metric for distance estimation. However, the hybrid distance is not defined on abstract objects as they do not have structure. Hence the medoids are used as cluster representatives during the post pruning step to refine the results produced by the ACO.

Scattering of a document on a 2D Grid

The size of the grid must be large enough to accommodate all the documents in its cells. In the variant proposed by the author a grid cell may accommodate multiple documents. A Two Dimensional grid is represented as a matrix with the number of grid cells equal to an integer multiple of P, where P is the total number of task relevant documents obtained through Document Score Estimation. Hence a grid is a 2 Dimensional array of vectors each containing a list of documents placed at a grid cell position <x,y>. The grid is implemented in Java using an appropriate instance of 'Multi-map Class' as it supports multi-dimensional indexing for storing list of values. Initially each document is assigned a random position on a 2D grid, which will be moved to the appropriate grid positions by the ants in the course of ACO clustering algorithm using hybrid distance metric which has been estimated in the earlier phase, as indicated in the previous chapter. For a given population to be clustered, larger grids are prone to form more number of clusters.

To maintain the details of all the P-relevant documents that are placed on the grid, a data structure in the form of an array is maintained for every document di. This array contains the information about a document placed on the grid regarding the position of the document and also the Boolean statusthat indicates whether a document is processed or not. Hence this array of P- structures contains details :<x-coordinate, y-coordinate, status> for each document. This array will be updated with its associated details of every document whenever a document is picked up or dropped down at a grid position by an ant which will be explained in the next steps of the process.

Positioning of an ant

An ant $a_i$is placed on a 2D grid following the similar procedure of placing a document on the grid. As proposed in [9] each ant reaches the documents in its neighborhood.As the documents are distributed randomly on the grid, ants can pick up and drop them in the grid cells depending upon the similarity between documents. The author proposed a variant of ant type as the ants are winged ants and hop over the 2D grid cells for picking up and dropping documents from/at appropriate neighborhoods based on document similarity. An ant searches for a document in its neighborhood defined by its surrounding 3x3 that is 9 neighboring cells including the grid cell occupied by the ant. The process of picking up and dropping down a document is explained in the next sections.

Picking a document

A document could be picked up by an ant only if an ant is not carrying any documents. If an ant $a_k$is not carrying a document, then estimate f(di) for each document in its 3x3 neighborhood using the grid position of a document. Divide the scale of [0, 1] to each document proportionate to its probability of picking up Pp(di) as defined below:

$$P_p(di) = \left(\frac{k_1}{k_1 + f(di)}\right)^2 \qquad (6)$$

Where Pp =probability of picking a document di

k1 = 0.1(user defined threshold)

f(di)= local average similarity of di to the documents in its 3x3 neighboring cells, N(di) on the 2D grid.

f(di) is calculated using the following formula:

$$f(di) = \begin{cases} \frac{1}{s^2} \sum_{d_j \in N(di)} 1 - \frac{h(d_i,\ d_j)}{\alpha}, if\ f > 0 \\ 0,\ else \end{cases} \qquad (7)$$

s is fixed to a value 3 as the neighboring 3x3 grid cells around di define its locality.

α is a scale factor that separates the documents whose hybrid distance is beyond α to contribute negatively while those that are within α distance contributing positively to the average similarity of di , f(di). In this implementation α is taken as 0.5.

The document that is having higher probability has higher chance to be picked up if it is not already processed. It

follows that the document has to be removed from the appropriate vector of the 2D grid and also by updating the status of a document in the array of P-structures, which maintains the Boolean status of document processing. Each ant moves randomly on the grid, picks up a neighboring document di with probability Pp(di) and moves with the document di to a different position on the grid searching for a grid cell with similar neighboring grid cells so as to drop the document where the probability of dropping is more. Estimation of the dropping probability is explained in the next section.

Put down a document

An ant ak moves with a document di in its neighbourhood until it finds an appropriate position to drop the document dj. An ant drops a document i near a document j depending upon the similarity between document pair (i,j) using hybrid distance matrix calculated in the pre-processing phase of the framework in previous chapter. The dropping probability of a document is defined in terms of f(di) which is the local average similarity of di using the following formula:

$$P_{jd}(di) = \begin{cases} 2f(di), iff\ (di) < k_2 \\ 1,\ iff \geq 0 \end{cases} \quad (8)$$

Wherek2 is also user defined threshold with a value 0.15.

Once the document is dropped in its appropriate neighbourhood based on the probability, and the corresponding entry in the P- structures is updated so as to reflect the new document position.

The entire process of clustering using Ant Colony Optimization algorithm is described as follows:

Randomly distribute P documents d1,…dP on the grid G
Place M ants a1,..aM on the grid randomly
For T=1 to n do
For all ants akϵ {a1,..aM} do
If an ant ak does not carry document //Picking up a document
then
Check if there is a non-null list of documents at position(ak) then goto step 6 else
Repeat
Move the ant ak randomly to a new position and Check if there is a non-null list of documents at position(ak)
Until non-null list is found at position(ak)
For all documents in ak'sneighbourhood
Estimate f(di) and Pp(di)
Ant ak collects document di according to the probability Pp(di)
Reposition the ant ak randomly
Endif
If ant ak carries document di //Put down document
then
a. for(j=1: j<5:j++)
Estimate fj(di) and Pd(di) for jthneighbourhood reached by ant ak in jth attempt
Ant ak drops document di in jthneighbourhood with a probability Pjd(di)
Endif
Move ak to a new grid position unoccupied by another ant
End for
End for

The algorithm ends after a predefined number of iterations, resulting in a large number of clusters at several grid cells which will be merged as described below.

Cluster Formation by Merging

Clusters are formed after every document in the grid cells has been picked up/ dropped down depending on the calculation of function f(di). That means initial positions of the documents on the grid cells will be updated. After the repositioning of the documents is done, the new positions of the documents are merged with their associated neighbouring grid cells to form clusters. Each non-zero vector of the 2D grid represents a cluster at the beginning. Ultimately, a cluster is represented by a list of joinable grid points represented by a list of positions. The length of the vector gives the cardinality of a cluster while the list of documents it contains are the members of a cluster defined by each vector of significant length in the 2D array of vectors. The 2D grid cells are now processed by joining two clusters if they share a pair of joinable grid cells, indicated by a common coordinate in the pair of grid cells with the other coordinate differing by 1. Once the clusters are identified, the documents contained in each cluster and the distance between them are used to find the medoid of the cluster which is the document that is centrally located in the hybrid document space. A document di is centrally located in a cluster if ∑HD(di,dj) is minimum, where di,dj are confined to documents of the cluster,C.

The last stage of clustering is done by identifying the medoids of each cluster and declaring them as cluster representatives. All documents are distributed to their closest cluster representative based on hybrid distance as a post-pruning step to take care of those documents which were not picked up by any ants and hence left in their original random position. The process of post-pruning is detailed in the next section.

Post pruning

A cluster may contain some non-members in it as the ACO is executed limited number of iterations. The post pruning step has to identify such members and redistribute them into appropriate clusters. The following steps are executed:

Once the members of clusters are identified, the document whose distance to the rest of the documents of the cluster C, is less is found. This is measured using the formula given below:

$$M(C) = argmin_{di\epsilon C} \sum_{dj\epsilon C} HD(di,,dj) \quad (9)$$

It may be noted that each of these medoids (cluster representatives) is the documents that belong to D and hence the hybrid distances to these mediods to every other document is pre-calculated and available in hybrid distance matrix. Based on these distances each document can be made a member of the cluster with most similar medoid without requiring any additional computation.

Finding Right number of Clusters

A distinct feature of ACO based document clustering is that it has the capacity to automatically find the number of clusters for a given dataset. If the 2D grid is sufficiently sparse while accommodating the data points to be clustered it can successfully find the correct number of clusters. The table 2 depicts the number of clusters formed for various grid distributions; the grid occupancies are varied starting from dense with the number of grid cells limited to two times the number of documents, gradually increasing to an extent making it very sparse with hundred times the number of documents. It is evident from the table that the number of clusters increased up to the number of groups in the dataset and remained constant even if the grid is made very sparse. Dense grid distributions resulted in lesser number of clusters due to insufficient spacing on the 2D grid. The same observations were made when the ACO was applied on 10,000 document set, which were reported in table 7. Hence it may be concluded that the grid size should be about 10 times the number of data

points for obtaining appropriate clustering solution and larger grid will not spoil the quality of clustering which becomes evident in the following sections.

Evaluation of Cluster Solution

The obtained clusters are evaluated using the standard cluster efficiency measures such as Precision, Recall and F-measure.

Precision of a clustering solution GP, is the weighted sum of precision /purity IP(Ci) of individual clusters Ci. Here the weights are proportionate to the cluster cardinalities.

GP is estimated with the help of the following formula:

$$GP = \sum_{i=1}^{k} \frac{|C_i|}{P} IP(C_i) \quad (10)$$

Where P is the total number of plausibly relevant documents.

The precision of individual clusters IP(Ci) is calculated using the formula given below:

$$IP(Ci) = \frac{Numberofelementsinmajorityclasse \quad Ci}{|Ci|} \quad (11)$$

Recall of the cluster solution, GR is calculated as a weighted sum of recall of individual classes, IR(CLj). The weights are proportionate to the cardinalities of classes.

GR is estimated using the formula given below:

$$GR = \sum_{j=1}^{N} \frac{|CL_j|}{P} IR(CL_j) \quad (12)$$

The recall of individual classes is calculated as follows:

$$IR(CLj) = \frac{Numberofdocuments \quad frommajorityclasse \quad C_i}{|CL_j|} \quad (13)$$

F-measure is the harmonic mean of precision and recall of the clustering solution given by the following formula:

$$F\text{-measure} = \frac{2*GP*GR}{GP+GR} \quad (14)$$

The main aim of this section is to present the details of process of grouping documents as per user's interest more effectively by reducing the document corpora to P-plausibly relevant documents forming the task relevant document set and extracting inherent patterns of task relevant documents through clustering in support of focused search with respect to the user's information requirement. In other words, each cluster contains a set of documents with a common theme and hence the problem of searching in the large document corpora is reduced to finding an appropriate cluster and confining the search to its members whose cardinality is much smaller. The process of document retrieval from the resembling cluster is explained in the next chapter. The efficiency of this algorithm has been proven experimentally and the results are discussed in detail in the next section of this paper.

II. EXPERIMENTAL RESULTS

The algorithm is implemented on the data consisting of 10000 documents gathered from the standard data source Wikipedia[12]. The experimental results clearly indicate that the proposed hybrid method of ACO followed by post-pruning of the clusters proved better results when compared to the traditional algorithms in terms of precision, recall and F-measure. Experiments were also conducted to observe the time taken for clustering which is shown in the tables and the same is depicted using the values plotted as graphs.

The most important criterion of efficiency of an algorithm is the time taken for running an algorithm. An algorithm is considered to be efficient when it produces the desired output in less time. The proposed clustering algorithm using Ant Colony Optimization followed by post-pruning of the clusters is observed to be efficient compared to K-medoids clustering algorithm which is evident from the table given below:

Table I. Time taken for clustering using k-medoids and the proposed algorithm

Grid Size	Number of Clusters	Time taken by K-medoids in ms	Time taken by proposed algorithm in ms
2P	2	132	132
3P	3	139	125
4P	4	146	114
5P	5	149	100
6P	6	154	92
7P	7	158	84
8P	7	162	79
9P	8	179	71
10P	8	183	63
12P	9	189	59
13P	10	194	58
15P	10	198	57.9
18P	10	204	58.1
20P	10	209	57.9

The graph below shows that ACO combined with post pruning of the cluster results in terms of time taken in mille seconds against varying grid size.

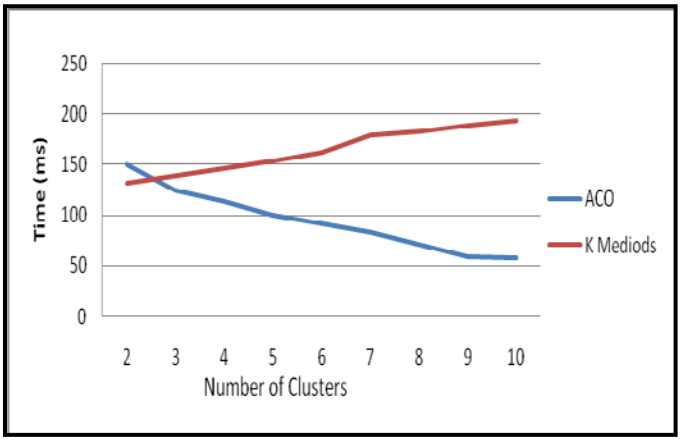

Figure 2. Time in milliseconds taken for clustering

All the above experimental results show that the proposed ACO algorithm followed by post pruning is proved to be better and efficient through the clustering evaluation methods used in this framework.

The proposed algorithm is also proved to be efficient by conducting experiments through performing clustering using traditional algorithm, k-medoids. The results obtained over k-medoids are compared with ACO algorithm performed using Euclidean distance and also ACO algorithm run through hybrid distance are tabulated and shown graphically below. The F-measure of the clustering solution is estimated and the comparison results are tabulated in the table II as shown below:

Table II. Comparison of clustering solution using F-measure obtained for k-medoids and ACO using different distance metrics.

Number of clusters	Fmeasure for k-medoids	Fmeasure for Aco with Euclidean distance	Fmeasure for Aco with hybrid distance
2	0.14	0.14	0.175
3	0.178	0.18	0.2592
4	0.198	0.2	0.3276
5	0.224	0.23	0.3807
6	0.261	0.27	0.4705
7	0.284	0.301	0.59549
8	0.301	0.334	0.67046
9	0.328	0.4054	0.703313
10	0.364	0.4297	0.71202
20	0.397	0.431	Does not arise

The above tabulated results are also visualized for better understanding in the form of a graph in figure 3 presenting number of clusters on x-axis and rate of F-measure on y-axis.

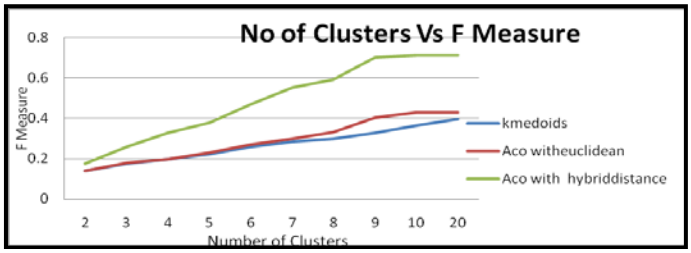

Figure 3. Comparison of various clustering algorithms with respect to F-measure

Thus the experimental results of clustering prove that the proposed clustering algorithm using hybrid distance in this framework is efficient through all the evaluation metrics that are considered. The obtained clusters are given as input to the next phase of this framework for ranking the document used for document retrieval in response to the given user's query. The process of ranking and retrieving relevant document is detailed in the later part of the paper.

In this framework information required is retrieved in the form of relevant documents from the document collection. The data corpora used in this framework is very big in cardinality consisting of 10000 documents and hence retrieving document to fulfil user's need requires searching over the entire corpora resulting in more search time. In order to avoid this and limit the search space, this research work aims at searching the required relevant document against the user query from the resembling cluster. The process of identifying resembling cluster and searching the details of cluster at micro-level to reach the resembling document with respect to the user's query is explained in the upcoming sections of this framework. The input of this phase of framework is the output obtained after performing clustering at two different stages - ACO using hybrid distance followed by pruning the clusters using medoids as cluster representatives

Each cluster contains semantically related group of documents on a specific topic and hence its centroid, referred to as 'Cluster Vector' captures the theme/topic shared by all the members of the cluster. In other words, each component of the 'Cluster Vector' corresponds to the weightage or relevance of a term. The resemblance of a cluster is measured as the summation of the weightages of the query terms. Thus the 'Cluster Vector' aids in finding the resemblance of a cluster to the user query.

The cluster that is having higher resemblance is identified and is processed in detail at micro-level. The members of the resembling cluster are processed by estimating the resemblance of each document with the summation of tf-idf of the terms corresponding to the query terms.

Now, the documents in the resembling cluster are sorted in decreasing order of their resemblance to the query terms and to generate ranking list of documents such thatthe document that exhibits higher relevance is placed at an earlier position in the ranking list.

Relevance Ranking

This phase of frame work aims at ranking the most appropriate collection of documents resembling the user query such that the most relevant documentsare placed at the top of the ranked list. These documents are sorted in decreasing order of their resemblance and ranked accordingly. A finite number of documents that exhibit higher relevance i.e, present at the top (top ranked) are retrieved, in response to the user's

requirement. The quality of ranking is measured in terms of Mean Average Precision for different sizes of ranking lists asthis measure imposes higher penalty for False Positives placed at the top positions of the ranking list compared to False Positives placed at later stages in the list.

The quality of relevance ranking is measured in terms of Average Precision (AP) which reflects both precision and recall capabilities of a ranking framework. Average Precision is defined as the mean of precision scores obtained at various stages in the ranking list in response to a given user query. AP is expressed as follows

$$AP = \frac{\sum_{k=1}^{n}(P(k)*rel(k))}{Numberofrelevantdocuments} \qquad (15)$$

Where P(k) is precision at kth position in the ranking list, which indicates the fraction of relevant documents among the top-k positions in the ranking list.

Rel(k)is the binary indicator which is equal to 1, if the document at kth position of the ranking list is relevant and 0 otherwise.

Average Precision is a monotonically increasing function in the length of ranking list. Average Precision varies in between 0 and 1.

Mean Average Precision(MAP) is the mean of the average precisions estimatedfor multiple queries of a given length.

For each query of a given length varying from 1 to 5 terms, the framework is applied for generating relevance ranking. Relevance Feedback is obtained from a pool of 10 users for each and every query and separate sets of relevant documents for each query are maintained. For a given query length, the Average Precision is estimated at stages 5, 10, 15, 20 and 25 for each query and the mean of the average precisions is determined and recorded in the table III. In order to compare the effectiveness of the framework for information retrieval, the documents are retrieved and ranked according to their relevance to queries from the whole corpus without screening using the framework and Mean Average Precisions are estimated using similar procedure. The Mean Average Precisions for queries of different lengths are presented in table 3.Experiments were performed on the entire document set as well as the documents residing in the cluster obtained through the proposed framework. The quality of ranking is measured in terms of Mean Average precision and the corresponding results were tabulated in the table III presented below. It is very much evident from the table that better ranking is observed with the algorithm defined in the framework after the screening is performed. The same has been presented in the figure 4.

Table III. Comparison of Mean Average Precision with screening and without screening

No Of Terms in Query	No of Documents Ranked	MAP Without Screening	MAP With Screening
1	5	0.26	0.31666
	10	0.31086	0.541766
	15	0.41278	0.68792
	20	0.53154	0.750456
	25	0.590383	0.750456
2	5	0.348674	0.4324
	10	0.456666	0.612403
	15	0.527786	0.69866
	20	0.576679	0.79766
	25	0.586679	0.79766
3	5	0.404486	0.47247
	10	0.507826	0.67346
	15	0.554255	0.751667
	20	0.626666	0.84793
	25	0.626666	0.84793
4	5	0.413074	0.55644
	10	0.544455	0.69356
	15	0.606616	0.82193
	20	0.687245	0.90347
	25	0.687245	0.90347
5	5	0.443075	0.57309
	10	0.616666	0.72863
	15	0.66799	0.85264
	20	0.741666	0.940437
	25	0.741666	0.940437

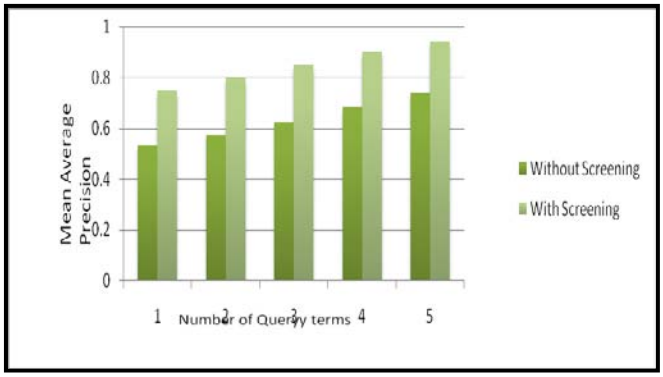

Figure 4. Barchart for increasing precision with increasing query terms

User's Information requirement is expressed more precisely by elaborating a query with more number of terms which leads to possibly better precision. Figure 3 presents evidence in tune with the above expectation using the framework. Mean Average Precisions for queries containing single term, two terms, three terms, four terms and five terms with and without screening are presented as bars and it may be observed that better Mean Average Precisions are obtained using the framework for screening the document corpus before ranking in response to the lengthier queries.

Hence, the proposed framework is successful in retrieving the most relevant document from the corpora for a given user query. It achieves the same in a much less time as is depicted in table 3 by performing the search in the subset of the corpora instead of searching the entire corpus.

III. CONCLUSIONS AND FUTURE WORK

The research work proposed and developed in this thesis deals with textual documents giving due weightage for frequency occurrence of terms belonging to vocabulary in the overall document for macro-level analysis and parts of the documents for micro-level analysis while estimating the hybrid distance between a pair of documents. This work can be extended for dealing with rich text documents possibly containing diagrams, tables, graphs etc., by appropriately extending the author proposed hybrid distance estimation methodology.

The proposed framework can be appropriately modified to handle incremental updates to the document corpora. Another possible extension suggests that in the context of Interactive Information Retrieval through small mobile devices where a shorter ranking list is preferred depending on the response given by the user to the top most portion of the ranking list generated from a single cluster the next chunk of document listing, can either be made as a continuation from the same cluster or the top most documents from the next best cluster.

IV. REFERENCES

[1] Amir Karshenas, KamilDimiller, 'PIRS: An Information Retrieval System based on Vector Space Model', International Symposium on Computer and Information Sciences, IEEE explore, 1-4, 2008.

[2] Zhuo Chen, Qing-Chun Meng, 'An Incremental Clustering Algorithm based on Swarm Intelligence Theory', International Conference on machine Learning and Cybernetics, IEEE Explore, Volume 3, 1768-1772, 2004.

[3] Marco Derigo, Vittorio Maniezzo, Alberto Colorni, 'The Ant System: Optimization by a Colony of Cooperative agents', IEEE Transactions on Systems, Man and Cybernetics- Part B, Vol.26, No.1, pp. 1-13, 1996

[4] Marco Derigo, Vittorio Maniezzo, Alberto Colorni, 'Distributed Optimization by Ant Colonies', Proceedings of 1st European Conference on Artificial Life, Elsevier,pp. 134-142, 1991.

[5] Marco Derigo, Vittorio Maniezzo, Alberto Colorni,' An Investigation on some properties of an Ant Algorithm', Proceedings of Parallel Problem Solving From Nature Conference, Elsevier Publishing, 509-520, 1992.

[6] Y.Kao, K.Cheng, 'An ACO based clustering algorithm', Proceedings of 5th International Workshop on Ant Colony Optimization and Swarm Intelligence, Vol. 4150, pp. 340-347. Changsheng Zhang, Mengli Zhu, Bin Zhang, 'An improved ant-based clustering algorithm', 8th International Conference on Natural Computation, IEEE, pp. 749- 752.

[7] Changsheng Zhang, Mengli Zhu, Bin Zhang, 'An improved ant-based clustering algorithm', 8th International Conference on Natural Computation, IEEE, pp. 749- 752.

[8] A. Vizine, L.N. de Castro, R.R.Gudwin, 'Towards Improving Clustering Ants: An Adaptive Clustering Algorithm', Informatica Journal, Vol.29, pp. 143-154, 2005.

[9] Lukasz Machnik, "ACO based document clustering method," Technical report ,Annales UMCS Informatica AI 3, pp 315-323, 2005.

[10] G. Salton, C. Buckley, 'Term Weighting Approaches in Automatic Text Retrieval', Journal of Information Processing and Management, Vol.24, No. 5, pp.513-523, 1988.

[11] VagelisHristidis, Yuheng Hu, Panagiotis.G, Ipeirotis, 'Relevance Based Retrieval on Hidden-Web Text Databases without Ranking Support', IEEE Transactions on Knowledge and Data Engineering, Vol. 23, Issue No. 10, pp. 1555-1568, 2011.

[12] www.wikepedia.com

[13] Michael Steinbach, George Karypis, VipinKumar,'A Comparison of Document Clustering Techniques'Department of Computer Science and Engineering, University of Minnesota ,Technical Report #00-034.,Feb 2001.

[14] Yizhou Sun, Hongbo Deng, Jiawei Han, 'Probabilistic Models for Text Mining', Chapter 8, Minin Text Data, Pearson Eduation, 2008.