Multi-Objective Optimization for Clustering of Medical Publications

Asif Ekbal **Sriparna Saha**
Indian Institute of Technology
Patna, Bihar, India
asif@iitp.ac.in
sriparna@iitp.ac.in

Diego Mollá
Department of Computing
Macquarie University, Sydney
NSW 2109, Australia
diego.molla-aliod@mq.edu.au

K Ravikumar
Indian Institute of Technology
Patna, Bihar, India
ravi.mc12@iitp.ac.in

Abstract

Clustering the results of a search can help a multi-document summarizer present a summary for evidence based medicine (EBM). In this work, we introduce a clustering technique that is based on multi-objective (MOO) optimization. MOO is a technique that shows promise in the areas of machine learning and natural language processing. In our approach we show how MOO based semi-supervised clustering technique can be effectively used for EBM.

1 Introduction

Evidence Based Medicine (EBM) urges the medical doctor to incorporate the latest clinical evidence available at point of care (Sackett et al., 1996). However, the amount of published clinical evidence is enormous. PubMed,[1] for example, indexes over 23 million citations, and the amount is growing every day. There are systematic reviews such as Cochrane's reviews that distill and summarize the information relevant to a particular topic, but often the doctor needs to access the primary literature, especially for cases that are rather infrequent and do not have systematic reviews dedicated to them, when dealing with particular segments of the population, or when the patient has simultaneous conditions ("comorbidity"). A search to PubMed can easily return hundreds of results, and finding specific information from that sea of information is time-consuming.

To help the doctor's need to find the evidence, it has been proposed to cluster the search results according to the different topics present in the clinical answer (Shash and Mollá, 2013). The motivation for this is that answers to a clinical question usually have several distinct parts, each of which is backed by a distinct set of published evidence. For example, as shown in Figure 1, the documents that answer the clinical inquiry *which treatments work best for hemorrhoids?* published in the Journal of Family Practice[2] can be grouped into three clusters, one for each suggested treatment (excision, hemorrhoidectomy, rubber band ligation).

Which treatments work best for hemorrhoids?

1. Excision is the most effective treatment for thrombosed external hemorrhoids. [11289288] [12972967] [15486746]
2. For prolapsed internal hemorrhoids, the best definitive treatment is traditional hemorrhoidectomy. [17054255] [17380367]
3. Of nonoperative techniques, rubber band ligation produces the lowest rate of recurrence. [1442682] [16252313] [16235372]

Figure 1: PubMed IDs of documents relevant to the answer to a clinical question.

We therefore propose to cluster all the documents relevant to a clinical query into clusters. Given a collection of clinical questions, the documents of each question represent a separate clustering task. In this paper, we present a method that uses multi-objective optimization techniques to cluster the results.

Section 2 gives a brief survey of clustering in general and within EBM. Section 3 introduces the general framework for the multi-objective optimization techniques that we use. Section 4 details the particular approach that we use to integrate multi-objective optimization techniques for clustering. Section 5 presents and discuss the results, and section 6 concludes this paper.

2 Brief Survey of Clustering

Document clustering is an unsupervised machine learning task that focuses on grouping similar doc-

[1] http://www.ncbi.nlm.nih.gov/pubmed

[2] http://www.jfponline.com

Asif Ekbal, Sriparna Saha, Diego Molla and K Ravikumar. 2013. Multi-Objective Optimization for Clustering of Medical Publications. In *Proceedings of Australasian Language Technology Association Workshop*, pages 53−61.

uments into clusters (Andrews and Fox, 2007). It has been used in a wide range of tasks such as Web search (Di Marco and Navigli, 2013), topic detection and tracking (Rajaraman and Tan, 2001), training data expansion for supervised classification (Karystinos and Pados, 2000), and multi-document summarization (Wang et al., 2008).

Document clustering has also been used within the domain of EBM. For example, Pratt and Fagan (2000) clustered search results corresponding to a user query. Lin and Demner-Fushman (2007) grouped MEDLINE citations into clusters based on interventions extracted from the document abstracts. Lin et al. (2007) used *K*-Means clustering to group PubMed query results. And Shash and Mollá (2013) used *K*-Means clustering to recover the original clusters used to determine the references relevant to clinical queries.

3 Formulation of Clustering as a Multi-objective Optimization Problem

Most of the existing clustering techniques are based on a single criterion which reflects a single measure of goodness of a partitioning. However, a single cluster quality measure is seldom equally applicable for different kinds of data sets with different characteristics. Hence, it may become necessary to simultaneously optimize several cluster quality measures that can capture different data characteristics. In order to achieve this, the problem of clustering a data set has been posed as one of multiobjective optimization (MOO) (Deb, 2001) in literature. Therefore, the application of sophisticated metaheuristic multiobjective optimization techniques seems appropriate and natural.

Determining the appropriate number of clusters from a given data set is an important consideration in clustering. For this purpose, and also to validate the obtained partitioning, several cluster validity indices have been proposed in the literature. The measure of validity of the clusters should be such that it will be able to impose an ordering of the clusters in terms of their goodness. In the literature there exists many cluster validity indices, that can be grouped mainly in two types: external and internal. In external validity indices, the true partitioning information (provided by user) is utilized while validating a particular partition. But in unsupervised classification, it is often difficult to generate such information. Because of this reason, external validity indices are rarely used to validate partitionings. Some common examples of such indices include *Minkowski score*s (Jiang et al., 2004) and *F-measures* (Saha and Bandyopadhyay, 2013). Internal validity indices rely on the intrinsic structure of the data. Most of the internal validity indices quantify how good a particular partitioning is in terms of the compactness and separation between clusters:

Compactness: This type of indices measures the proximity among the various elements of the cluster. One of the commonly used measures for compactness is the variance.

Separability: This particular type of indices is used in order to differentiate between two clusters. Distance between two cluster centroids is a commonly used measure of separability. This measure is easy to compute and can detect hyperspherical-shaped clusters well.

Some well-known internal cluster validity indices are the BIC-index (Raftery, 1986), CH-index (Caliński and Harabasz, 1974), Silhouette-index (Rousseeuw, 1987), DB-index (Davies and Bouldin, 1979), Dunn-index (Dunn, 1973), XB-index (Xie and Beni, 1991), PS-index (Chou et al., 2002), and I-index (Maulik and Bandyopadhyay, 2002). Maulik and Bandyopadhyay (2002) show the effectiveness of I-index and XB-index compared to the other indices in determining the appropriate number of clusters from the data sets. Being guided by these observations we use these two cluster validity indices as the two objective functions in our proposed multiobjective clustering technique. However it is to be noted that the proposed algorithm is very general, and can be applicable with any sets of cluster validity indices. These objectives are not conflicting to each other, and their (I-index and XB-index) goals are to minimize cluster compactness and maximize cluster separation. But while XB-index maximizes minimum distance between any two cluster centroids, I-index maximizes maximum distance between any two cluster centroids. This difference helps them to determine different sets of clusters from a data set.

3.1 I-Index

The I-index (Maulik and Bandyopadhyay, 2002) is defined in the following equation:

$$I(K) = (\frac{1}{K} \times \frac{\mathcal{E}_1}{\mathcal{E}_{\mathcal{K}}} \times D_K)^p \tag{1}$$

where K is the number of clusters. Here

$$\mathcal{E}_{\mathcal{K}} = \sum_{k=1}^{K} \sum_{j=1}^{n_k} d_e(\overline{c}_k, \overline{x}_j^k) \tag{2}$$

and

$$D_K = \max_{i,j=1}^{K} d_e(\overline{c}_i, \overline{c}_j) \tag{3}$$

where $\overline{c}_j$ denotes the centroid of the jth cluster and $\overline{x}_j^k$ denotes the jth point of the kth cluster. The number n_k is the total number of points present in the kth cluster. The value of K for which I-index takes its maximum value is considered as the appropriate number of clusters.

The index I is a composition of three factors, namely $\frac{1}{K}$, $\frac{\mathcal{E}_1}{\mathcal{E}_{\mathcal{K}}}$ and D_K. The first factor attempts to reduce index I as the value of K is increased. The second factor is the ratio of $\mathcal{E}_1$ and $\mathcal{E}_{\mathcal{K}}$. While the former remains constant for a given data set, the later decreases with increase in K. Hence, because of this term, index I gradually increases as $\mathcal{E}_{\mathcal{K}}$ decreases. This, in turn, denotes that formation of more numbers of compact clusters would be encouraged. Finally, the third factor, D_K, measures the maximum separation between two clusters over all possible pairs of clusters. This increases proportionally with the value of K. However, the ultimate value of this factor can exceed the maximum separation between two points in the data set. Thus, the three factors are found to compete with and balance each other critically. The power p is used to control the contrast between the different cluster configurations. In this paper, we set the value of p to 2.

3.2 XB-Index

The second objective function used in the clustering algorithm is the XB-index. This is one of the widely used internal cluster validity indices in the literature. In 1991, Xie and Beni (1991) developed this cluster validity index (XB-index) which is again based on two properties: compactness and separation. As per the definitions the numerator quantifies the compactness of the partitioning while the denominator quantifies the separation between clusters. Separation is measured based on the Euclidean distance between the cluster centroids. In principle, in order to attain a good partitioning, the compactness value should be minimum and the separation should be maximum. Therefore, in order to obtain a desirable partitioning, the value of XB-index should be minimized after varying the number of clusters in the range, $k = 1, \ldots, K_{max}$. Let K cluster centroids be represented by $\overline{c}_i$ where $1 \leq i \leq K$ and $[u_{ij}]_{K \times n}$ denote the membership matrix for the data. Then the XB-index is defined by the following equation:

$$XB(K) = \frac{\sum_{i=1}^{K} \sum_{j=1}^{n} u_{ij}^2 \|\overline{x}_j - \overline{c}_i\|^2}{n(\min_{i \neq k} \|\overline{c}_i - \overline{c}_k\|^2)} \tag{4}$$

Thus the two objective functions used for clustering are $f_1 = I$ and $f_2 = \frac{1}{XB}$. The clustering algorithm will attempt to maximize these two indices.

3.3 Multi-Objective Optimization

Multi-objective optimization can be formally stated as follows: find the vector $\overline{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ of decision variables that simultaneously optimize M objective values

$$\{f_1(\overline{x}), f_2(\overline{x}), \ldots, f_M(\overline{x})\}$$

while satisfying user-defined constraints, if any.

An important concept in MOO is that of domination. Within the context of a maximization problem, a solution $\overline{x_i}$ is said to dominate $\overline{x_j}$ if $\forall k \in 1, 2, \ldots, M,\ f_k(\overline{x_i}) \geq f_k(\overline{x_j})$ and $\exists k \in 1, 2, \ldots, M,$ such that $f_k(\overline{x_i}) > f_k(\overline{x_j})$. Among a set of solutions P, the nondominated set of solutions P' are those that are not dominated by any member of the set P. The nondominated set of the entire search space S is called the globally Pareto-optimal set or Pareto front. In general, a MOO algorithm outputs a set of solutions not dominated by any solution encountered by it.

These notions can be illustrated by considering an optimization problem with two objective functions — say, f_1 and f_2 — with five different solutions, as shown in Figure 2. In this example, solutions 3 and 5 dominate all the other three solutions 1, 2 and 4; solutions 3 and 5 are non-dominating to each other, because whereas 5 is better than 3 with respect to f_1, 3 is better than 5 with respect to f_2. Therefore, the Pareto front is made of solutions 3 and 5.

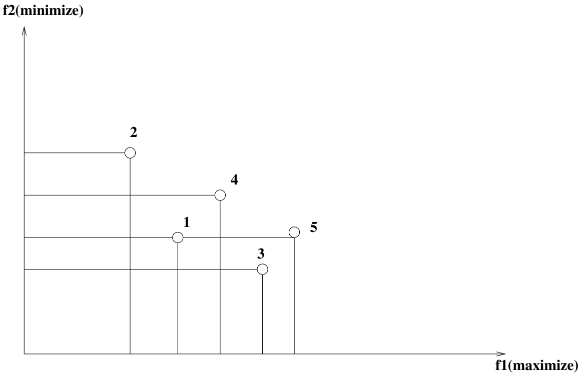

Figure 2: Example of dominance and Pareto optimal front.

4 Proposed Method of Multi-Objective Clustering

This section describes the multi-objective clustering technique, *AMOSA-clus*, in detail. This technique uses AMOSA (Bandyopadhyay et al., 2008) as the underlying optimization strategy. A short description of AMOSA is also provided in this section.

4.1 String Representation and Population Initialization

In *AMOSA-clus* clustering, centroid-based real-encoding is used. Here each member of the archive is encoded as a string that represents the coordinates of the centroids of the partitions. Each string has a different length. Let us assume string i represents the centroids of K_i clusters and the dimension of the data space is d, then the string has length l_i where $l_i = d * K_i$. For example, in the case of two-dimensional space, the string

$$< 12.3\ 1.4 \quad 22.1\ 0.01 \quad 0.0\ 15.3 \quad 10.2\ 7.5 >$$

represents four cluster centroids:

$$(12.3, 1.4), (22.1, 0.01), (0.0, 15.3), (10.2, 7.5)$$

An important point of string encoding is that each centroid is regarded to be indivisible. This means at the time of mutation if we will insert a new centroid all the dimensional values have to be inserted and if we want to delete a centroid all the dimensional values have to be deleted. The number of centroids, K_i, encoded in a string i is chosen randomly between two limits K_{min} and K_{max}. The value is determined using the following equation:

$$K_i = (rand() \text{mod}(K_{max} - 1)) + 2 \qquad (5)$$

Here, $rand()$ is a function returning a random integer number, and K_{max} is the upper-limit of the number of clusters. The minimum number of clusters is assumed to be 2. The number of whole clusters present in a particular string of archive can therefore vary in the range of two to K_{max}. The K_i cluster centroids represented in a string are some randomly selected distinct points from the data set.

4.2 Assignment of Points to Different Clusters and Objective Function Computations

The computation of the objective functions is done in two steps. The first step concerns with the assignment of n points (where n is the total number of points in the data set) to different clusters. In the second step, we compute our two cluster validity indices, XB-index (Xie and Beni, 1991) and I-index (Maulik and Bandyopadhyay, 2002), and use them as two objective functions of the string. Thereafter we simultaneously optimize the two objective functions using the search capability of AMOSA.

4.2.1 Assignment of Points to Different Clusters

In *AMOSA-clus*, the assignment of points to different clusters is done based on the minimum distance based criterion in a similar way as is done in an iteration of the *K*-means clustering algorithm. In particular, any point j is assigned to a cluster k whose centroid has the minimum distance to j. That is:

$$k = argmin_{i=1,...K} d(\overline{x}_j, \overline{c}_i) \qquad (6)$$

K denotes the total number of clusters, $\overline{x}_j$ is the jth data point, $\overline{c}_i$ is the centroid of the ith cluster and $d(\overline{x}_j, \overline{c}_i)$ denotes some distance measure between the data point $\overline{x}_j$ and cluster centroid $\overline{c}_i$.

After assigning all the points to different clusters, the cluster centroids represented in a particular string of the archive are updated by the average of the points which are in a single cluster:

$$\overline{c}_i = \frac{\sum_{j=1}^{n_i} (\overline{x}_j^i)}{n_i}, \quad 1 \leq i \leq K \qquad (7)$$

Where n_i is the number of points in cluster i and $\overline{x}_j^i$ is the jth point of the ith cluster.

4.3 Search Operators

As mentioned earlier the proposed clustering technique uses a multiobjective simulated annealing based technique as the underlying optimization technique. As a simulated annealing step, we need to introduce mutation operations. We introduce three:

Mutation 1: In this mutation each cluster centroid is changed by some small amount. The Laplacian distribution is used in order to generate some completely random numbers. Here each cluster centroid represented in a string is modified with a random variable which is drawn using a Laplacian distribution,

$$p(\epsilon) \propto e^{-\frac{|\epsilon-\mu|}{\delta}}$$

The magnitude of perturbation is measured using the scaling factor δ and μ is the old value at the position which is to be mutated. The scaling factor δ is generally set equal to 1.0. By using the Laplacian distribution a value near the old value is generated and the old value is replaced with the newly generated value. This is applied individually to all the dimensions of a particular centroid if it is selected for mutation.

Mutation 2: This mutation operation is used to reduce the size of the string. A cluster centroid is generated at random and selected to be deleted from the string. This is done to decrease the number of cluster centroids encoded in the string by 1. Cluster centroids are considered to be indivisible. This means as a result of deleting a particular cluster centroid, all the dimensional values are removed.

Mutation 3: This mutation is for incrementing the number of clusters by 1. One new centroid is inserted in the string, and so the number of cluster centroids encoded in the string is incremented by 1. As the cluster centroids are indivisible, all the dimensional values of the centroid, selected randomly, are inserted into the string.

For example, let the string $< 3.5\ 1.5 \quad 2.1\ 4.9 \quad 1.6\ 1.2 >$ represent three cluster centroids in a 2-d plane $(3.5, 1.5)$, $(2.1, 4.9)$, and $(1.6, 1.2)$.

1. For mutation type 1, let position 2 be selected randomly. Then, each dimension of $(2.1, 4.9)$ will be changed by some values generated using the Laplacian distribution.

2. If mutation type 2 is selected, a centroid will be removed from the string. Let centroid 3 be selected for deletion. Then, after deletion, the string will look like $< 3.5\ 1.5 \quad 2.1\ 4.9 >$.

3. In case of third mutation, a new centroid will be added to the string. Let the randomly chosen point from the data set to be added to the string be $(9.7, 2.5)$. After inclusion of this centroid, the string looks like $< 3.5\ 1.5 \quad 2.1\ 4.9 \quad 1.6\ 1.2 \quad 9.7\ 2.5 >$.

In order to generate a new string any one of the above-mentioned mutation types is applied to each string. We have associated equal probability with each of these mutation operations. Thus in 33% cases mutation 1, in 33% cases mutation 2 and in 33% cases mutation 3 take place.

4.4 Selecting a Single Solution from the Pareto Optimal Front

Any multi-objective optimization technique produces a set of non-dominated solutions on its final Pareto optimal front (Deb, 2001). Each of these non-dominated solutions corresponds to a complete assignment of clusters to the data set. In the absence of additional information, any of those solutions can be selected as the optimal solution. But sometimes the user can have labelled information for some portions of the dataset. In this section we describe a process of semi-supervised clustering where, for every question, a portion of the documents are already clustered. This could happen, for example, when someone wants to update some known evidence with further evidence gathered via a document search process. The known information can be used to select one of the non-dominated solutions from the final Pareto front.

In our experiments, we use cluster entropy to determine the best solution from the Pareto front. Cluster entropy is calculated based on the cluster precision, that is the ratio of elements retrieved from a particular source cluster. Thus, to compute the entropy of cluster i, we first determine how many data points from each source cluster j appear in cluster i, relative to the size of cluster i:

$$p_{ij} = \frac{m_{i,j}}{m_i} \qquad (8)$$

Then the entropy of cluster i is:

$$Entropy(i) = -\sum_{j} p_{i,j} \times log_2 p_{i,j} \quad (9)$$

The entropy measure of the clusters generated for a particular data set is the weighted sum of the entropies of all clusters for that data set. Here the weight is the ratio of the cluster size relative to the total number of data points present in the data set.

For every non-dominated solution, the entropy values of the training set are computed, and the solution with lowest (best) entropy is selected. For the results presented in this paper we have chosen a training set of 10% of the total data points.

Let us take an example. Suppose that we have four questions, each one with five documents. The set of documents is:

$$S = \begin{array}{l} \{\{a,b,c,d,e\},\{f,g,h,i,j\}, \\ \{k,l,m,n,o\},\{p,q,r,s,t\}\} \end{array}$$

We apply the *AMOSA-clus* clustering technique on these four questions separately. For the sake of this example, for each question we select one document as the training set. Let us assume there is a total of N solutions on the final Pareto front. Based on each of these N solutions, we assign a class label to this training document. Now the entropy value is computed for this one document for each solution. The solution with *minimum entropy* value is selected as the optimal solution. Now the centers encoded in this solution are used to assign class labels to the remaining four documents. Next *AMOSA-clus* is applied on the second question and the same procedure is repeated to calculate the overall entropy for the second question. In this way the *AMOSA-clus* clustering technique is applied for all the questions and the same procedures are repeated to compute the final results.

4.5 The SA Based MOO Algorithm: AMOSA

Archived multi-objective simulated annealing (AMOSA) (Bandyopadhyay et al., 2008) is an efficient MOO version of the simulated annealing (SA) algorithm. Simulated annealing is a search technique for solving difficult optimization problems, which is based on the principles of statistical mechanics (Kirkpatrick et al., 1983). Although the single objective version of SA has been quite popular, its utility in the multi-objective case was limited because of its search-from-a-point nature. Recently Bandyopadhyay et al. (2008) developed an efficient multi-objective version of SA called AMOSA that overcomes this limitation.

The AMOSA algorithm incorporates the concept of an archive where the non-dominated solutions seen so far are stored. Two limits are kept on the size of the archive: a hard or strict limit denoted by *HL*, and a soft limit denoted by *SL*. Given $\gamma > 1$, the algorithm begins with the initialization of a number $(\gamma \times SL)$ of solutions each of which represents a state in the search space. The multiple objective functions are computed. Each solution is refined by using simple hill-climbing and domination relation for a number of iterations. Thereafter the non-dominated solutions are stored in the archive until the size of the archive increases to *SL*. If the size of the archive exceeds *HL*, a single-linkage clustering scheme is used to reduce the size to *HL*. Then, one of the points is randomly selected from the archive. This is taken as the current-pt, or the initial solution, at temperature $T = Tmax$. The current-pt is perturbed/mutated to generate a new solution named new-pt, and its objective functions are computed. The domination status of the new-pt is checked with respect to the current-pt and the solutions in the archive. A new quantity called amount of domination, $\Delta dom(a,b)$ between two solutions a and b is defined as follows:

$$\Delta dom(a,b) = \prod_{i=1, f_i(a) \neq f_i(b)}^{M} \frac{f_i(a) - f_i(b)}{R_i} \quad (10)$$

where $f_i(a)$ and $f_i(b)$ are the ith objective values of the two solutions, R_i is the corresponding range of the objective function and M is the number of objective functions. Based on domination status different cases may arise viz., accept the (i) new-pt, (ii) current-pt, or, (iii) a solution from the archive. Again, in case of overflow of the archive, clustering is used to reduce its size to *HL*. The process is repeated *iter* times for each temperature that is annealed with a cooling rate of $\alpha(< 1)$ till the minimum temperature $Tmin$ is attained. The process thereafter stops, and the archive contains the final non-dominated solutions.

In order to reduce the temperature, we have used geometric cooling: $T_{k+1} = \alpha \times T_k$ where α is the cooling rate. We have used $\alpha = 0.9$ in the current paper. We use AMOSA as the underlying MOO technique in this work because of its improved performance over some other well-known MOO algorithms especially for three or more objectives (Bandyopadhyay et al., 2008).

5 Results

Below we present the results based on a random partition of 276 clinical questions from the corpus by Mollá and Santiago-Martínez (2011). Each question has an average of 5.89 documents. The corpus is based on the material from the Clinical Inquiries section of the Journal of Family Practice. The data set has information about the question, the answer, and the documents that are relevant to each part of the answer, as illustrated in the example of Figure 1. The documents of each of the answer parts determines a cluster. The *AMOSA-clus* clustering technique is therefore applied on each question individually. The average entropy value of all the questions is then calculated. The parameters of the *AMOSA-clus* clustering technique are as follows: *SL*=100 *HL*=50, *iter*=50, *Tmax*=100, *Tmin*=0.0001 and cooling rate $\alpha = 0.9$.

Table 1 compares the entropy results for clustering using *AMOSA-clus* with a fixed and variable number of clusters. We experimented with two cluster measures of document distance: Euclidean distance, and cosine distance. The cosine distance is computed as 1-cosine similarity. Strictly speaking this is not a distance metric but it is included to compare with the results presented by Shash and Mollá (2013), who reported optimal results by using *K*-means with this use of the cosine distance, and which we also include in the table as the baseline.[3] We include the Euclidean distance since this is the standard metric used for *K*-means clustering and is also reported by Shash and Mollá (2013). All the results reported in the table, included the *K*-means baseline, are based on the same partition of 276 questions from the corpus developed by Mollá and Santiago-Martínez (2011).

Each document is represented as a vector of *tf.idf* values based on stemmed and lowercased words, with stop words removed.

5.1 Finding the Number of Clusters

The training set includes information about the actual number of clusters. We have used this information to test *AMOSA-clus*' ability to determine the optimal number of clusters, by implementing two variants: *AMOSA-clus1* performs clustering by fixing the number of clusters to the number provided by the corpus, whereas *AMOSA-clus2* automatically determines the optimal number of clusters.

Table 2: Measure of the error of number of clusters of *AMOSA-clus2* and a number of popular methods.

Method	Error
AMOSA-clus2 Cosine	1.90
AMOSA-clus2 Euclidean	1.91
$k = 1$	3.91
$k = 2$	2.14
$k = 3$	2.38
$k = 4$	4.61
Rule of Thumb	2.56
Cover	1.98

AMOSA-clus2 is executed on each question by varying the number of clusters in a range between 2 and $\sqrt{n}$ where n is the number of documents per question, and using the above mentioned indices I-index and XB-index to determine the best solution. The average number of clusters identified by this procedure for each question is 2.51 and 2.34, respectively, with cosine and Euclidean distance measurements. The average number of clusters in the actual annotated set is 2.38. Since entropy is based on cluster precision, a larger number of clusters will naturally lead to a better value of entropy, reaching a perfect zero when there are as many clusters as documents. Consequently, we can only rely on the Euclidean metric (with average 2.34 clusters) to assess the efficacy of the automatic selection of number of clusters. We observe that the results of *AMOSA-clus2* using the Euclidean metric is slightly better than *AMOSA-clus1*, which gives some evidence that the proposed *AMOSA-clus2* technique to determine the number of clusters is promising.

Next we have compared the generated number of clusters with the known number of clusters using the mean of the squares of the errors:

$$error = \frac{\sum_i (target_i - predicted_i)^2}{\#\ of\ questions} \qquad (11)$$

Table 2 compares the error in the generation of numbers of clusters between *AMOSA-clus2* and a set of heuristics widely used in the literature: fixed number of clusters ($k = 1, 2, 3, 4$), the Rule

[3] Our baseline is a replication of the original paper's experiment and the results are different.

Table 1: Average Entropy values obtained by two variants of *AMOSA-clus* and a baseline *K*-means clustering technique for whole XML files; here *AMOSA-clus1*: *AMOSA-clus* with fixed number of clusters, *AMOSA-clus2*: *AMOSA-clus* with variable number of clusters, *K*-means: *K*-means with fixed number of clusters; best: entropy value of the solutions selected by the procedure described in Section 4.4; average: average entropy of all the solutions present on the final Pareto front.

Distance Measure	*AMOSA-clus1*		*AMOSA-clus2*		*K*-means (baseline)
	best	average	best	average	
Euclidean	0.190	0.249	0.177	0.235	0.240
Cosine	0.187	0.231	0.177	0.230	0.237

of Thumb ($k = \sqrt{n/2}$) (Mardia et al., 1979), and the cover method (Can and Esen A. Ozkarahan, 1990). We observe that the error of *AMOSA-clus2* is lowest in both distance measures, cosine and Euclidean. We conducted a Wilcoxon signed-rank test and observed that the differences in the squared errors between the *AMOSA-clus2* variants and the cover method are statistically significant.

5.2 Semi-supervised Setting

Each *AMOSA-clus1* and *AMOSA-clus2* has been run both in a semi-supervised setting and a fully unsupervised setting. In the semi-supervised setting, the information of 10% of the documents relevant to a question is used to select the best non-domimant solution from the Pareto front as described in Section 4.4. The entropy reported in the *best* column of Table 1 indicates the entropy values after disregarding the 10% documents used to select the solution. In the unsupervised setting, we report the average of all solutions of the Pareto front and is presented in the *average* column. We observe that the semi-supervised approach produces a better (lower) entropy, and a Wilcoxon signed-rank test reveals that the difference with respect to the baseline *K*-means clustering method is statistically significant. The results of the unsupervised setting also have a statistically significant difference with the baseline, though we can observe that the difference is much lesser and in one case it is worse.

6 Conclusions

We have presented a novel approach for clustering documents that is based on the use of multi-objective optimization (MOO), for the task of splitting the documents relevant to the answer of a clinical question into each of the answer parts. The MOO approach is based on a variant of Archived Multi-Objective Simulated Annealing (AMOSA) that we call *AMOSA-clus*, which uses cluster-based evaluation indices as the objectives to optimize. Even though the results do not show an improvement over a baseline of *K*-means reported in the literature, a semi-supervised variant shows an improvement over the baseline. Our experiments show the effectiveness of the use of MOO techniques for this clustering task in particular. Given the generality of the approach proposed, it is reasonably to conclude that these MOO techniques would be useful in a general clustering setting.

We have experimented with a variant that uses the known cluster numbers, and another variant that automatically determines the optimal number of clusters. The good results of the option with automatic number of clusters show the promising potential of this approach.

The improvement of results by using MOO techniques are highly encouraging. Further work can be done in several fronts. First of all, further experiments are required to improve the efficacy of the automatic selection of the number of clusters. Also, it is desirable to test whether *AMOSA-clus* improves the results in other clustering applications such as the ones briefly mentioned in Section 2. In our experiments we used the I and XB indices as the objective functions to optimise due to their general popularity. It would be interesting to test the use of other combinations of cluster validity indices, or even to build a MOO system that uses a larger selection of them.

Within the area of multi-document summarization, further work will focus on the determination of techniques of extraction or generation of topic labels that could be used for the generation of the final summaries.

References

Nicholas O. Andrews and Edward A. Fox. 2007. Recent Developments in Document Clustering. Technical report, Virginia Tech.

Sanghamitra Bandyopadhyay, Sriparna Saha, Ujjwal Maulik, and Kalyanmoy Deb. 2008. A simulated annealing based multi-objective optimization algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation*, 12(3):269–283.

R. B. Caliński and J. Harabasz. 1974. A dendrite method for cluster analysis. *Comm. in Stat.*, 3:1–27.

Fazli Can and Esen A. Ozkarahan. 1990. Concepts and Effectiveness of the Cover-Coefficient-Based Clustering Methodology for Text Databases. *ACM Transactions on Database Systems*, 15(4):483–517.

Chien-Hsing Chou, Mu-Chun Su, and Eugene Lai. 2002. Symmetry as a new measure for cluster validity. In *2nd WSEAS Int. Conf. on Scientific Computation and Soft Computing*, pages 209–213. Crete, Greece.

David L. Davies and Donald W. Bouldin. 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:224–227.

Kalyanmoy Deb. 2001. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, Ltd, England.

Antonio Di Marco and Roberto Navigli. 2013. Clustering and Diversifying Web Search Results with Graph-Based Word Sense Induction. *Computational Linguistics*, 39(3):709–754, November.

J. C. Dunn. 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57.

Daxin Jiang, Chun Tang, and Aidong Zhang. 2004. Cluster analysis for gene-expression data: A survey. *IEEE Trans. Knowledge Data Eng.*, 16:1370–1386.

G N Karystinos and D A Pados. 2000. On overfitting, generalization, and randomly expanded training sets. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 11(5):1050–7, January.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by simulated annealing. *Science*, 220:671–680.

Jimmy J. Lin and Dina Demner-Fushman. 2007. Semantic clustering of answers to clinical questions. In *AMIA Annual Symposium Proceedings*.

Yongjing Lin, Wenyuan Li, Keke Chen, and Ying Liu. 2007. A Document Clustering and Ranking System for Exploring {MEDLINE} Citations. *Journal of the American Medical Informatics Association*, 14(5):651–661.

Kanti V. Mardia, John T. Kent, and John M. Bibby. 1979. *Multivariate Analysis*. Academic Press, London.

Ujjwal Maulik and Sanghamitra Bandyopadhyay. 2002. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1650–1654.

Diego Mollá and Maria Elena Santiago-Martínez. 2011. Development of a Corpus for Evidence Based Medicine Summarisation. In *Proceedings of the Australasian Language Technology Workshop*.

Wanda Pratt and Lawrence Fagan. 2000. The Usefulness of Dynamically Categorizing Search Results. *Journal of the American Medical Informatics Association*, 7(6):605–617.

A. Raftery. 1986. A note on Bayes factors for log-linear contingency table models with vague prior information. *Journal of the Royal Statistical Society*, 48(2):249–250.

Kanagasabi Rajaraman and Ah-Hwee Tan. 2001. Topic Detection, Tracking, and Trend Analysis Using Self- Organizing Neural Networks. In *PAKDD '01 Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 102–107, London. Springer-Verlag.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comp App. Math*, 20:53–65.

David L Sackett, William M Rosenberg, Jamuir Gray, R Brian Haynes, and W Scott Richardson. 1996. Evidence Based Medicine: What it is and what it isn't. *BMJ*, 312(7023):71–72.

Sriparna Saha and Sanghamitra Bandyopadhyay. 2013. A generalized automatic clustering algorithm in a multiobjective framework. *Appl. Soft Comput.*, 13(1):89–108.

SF Shash and D Mollá. 2013. Clustering of Medical Publications for Evidence Based Medicine Summarisation. In *Artificial Intelligence in Medicine*, pages 305–309.

Dingding Wang, Shenghuo Zhu, Tao Li, Yun Chi, and Yihon Gong. 2008. Integrating Clustering and Multi-document Summarization to Improve Document Understanding. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1435–1436.

Xuanli Lisa Xie and Gerardo Beni. 1991. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:841–847.