CLUSTER ANALYSIS APPLIED TO EUROPEANA DATA

by

Esra Atescelik

In partial fulfillment of the requirements

for the degree of Master of Computer Science

Department of Computer Science

VU University Amsterdam

ABSTRACT

CLUSTER ANALYSIS APPLIED TO EUROPEANA DATA

Esra Atescelik

Department of Computer Science

VU University Amsterdam

e.atescelik@student.vu.nl

2014

The digital libraries and aggregators such as Europeana provide access to millions of Cultural Heritage objects (CHOs). Europeana is one of the libraries which does not maintain collection-level metadata. Europeana can cluster the objects that have common information with each other. Clustering would help handle the variety of search results on Europeana. It would help the users to find related items.

The purpose of one of our experiments is to see how we can cluster the objects from Europeana datasets. We also aim at finding the best way of clustering on Europeana metadata and the best parametric setting for clustering.

We apply various clustering methods on Europeana metadata and aim at proposing a clustering technique that is most appropriate to group Europeana Cultural Heritage objects (CHOs). In the experiments we evaluated the cluster results manually, on qualitative and quantitative level.

The results of experiments showed that it is difficult to define the best parametric setting and best clustering method only based on a number of experiments. However, we have shown a process to cluster Europeana objects which may be useful for Europeana.

CONTENTS

CHAPTER 1: INTRODUCTION

The digital libraries and aggregators such as Europeana provide access to millions of Cultural Heritage objects (CHOs). Europeana gives people free and open access to large number of digital books, paintings, films, sounds, museum objects and archives throughout Europe [1]. More than 2.300 institutions have contributed to Europeana such as the British Library in London, the Rijksmuseum in Amsterdam or the Kunsthistorisches Museum in Vienna. An important property of Europeana is that it connects the user to the original source of the material, the user can be sure about its reality, authenticity. People can also contribute to Europeana. For instance, they can upload their digitised items onto the Europeana1914-1918.eu site [1].

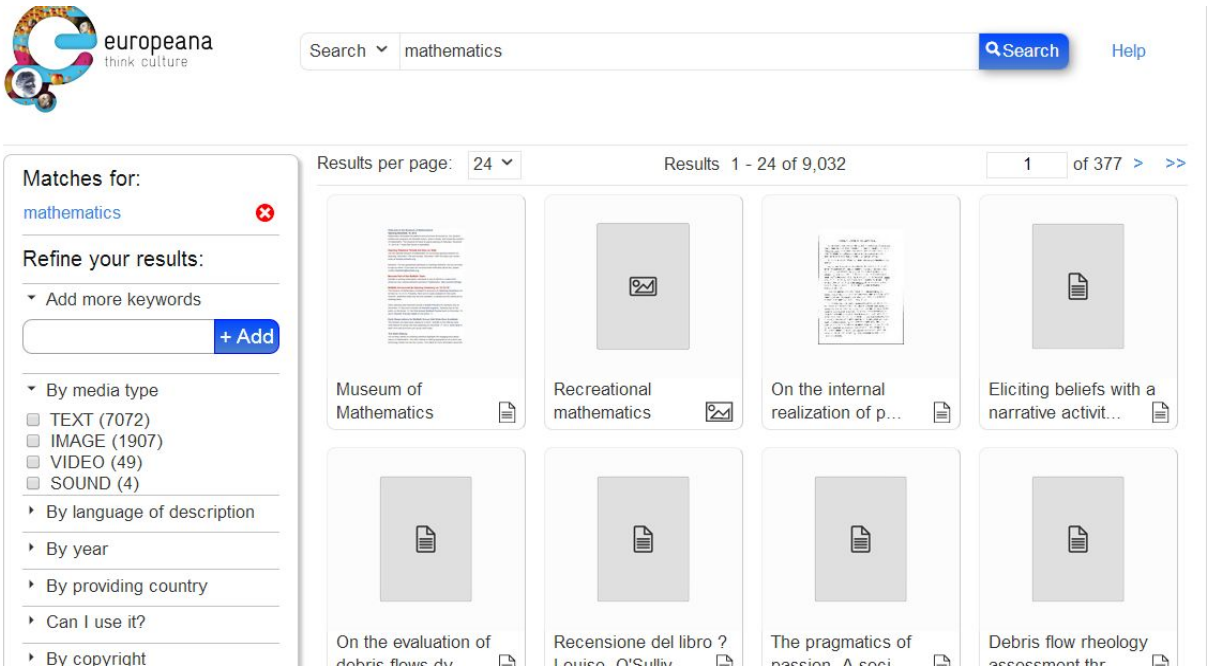
1.1. Problem Statement

While large sets of various collections of items are daily increasing they need to be managed and presented effectively for information access especially in the libraries, museums, and archives. For this reason, the use of collection-level descriptions has become a relevant topic in last years particularly because the digital libraries stand for a more heterogeneous manifestation than traditional libraries. We can gather digital resources into 'collections' [2]. A collection is defined [3] as "a group of objects gathered together for some intellectual, artistic, or curatorial purpose." An object is the resource being described [4], it can be physical (e.g. a book, a painting) or born-digital (e.g. a 3D model) [3].

One of the main current problems on Europeana is that Europeana does not maintain the collection-level metadata yet. Collection-level metadata [5] is metadata describing an aggregation of objects such as the photo album that contains a group of photographs. Europeana cannot use collection-level information to organize results and help users as [3] proposes. To achieve these goals, we want to find the best way of clustering applied on Europeana data.

Europeana can cluster the objects that have common information with each other. An object can be a painting, photo, picture, book, music, film, letter etc. For example, assume that we want to get all the objects by the subject 'mathematics' on Europeana.eu. In this case, we use 'mathematics' as search term. As a result, Europeana returns mainly objects about the subject 'mathematics', but also objects of which the title, and/or description include 'mathematics'. You can see an overview of search results in Figure 1. Clustering would help handle the variety of search results and provide better information on their context.

Figure 1: Search results of the keyword 'mathematics' on Europeana.eu.



Europeana could use the metadata on collection level:

- to give an overview of the subjects included by the Europeana collections
- to group objects belonging to the same collections and provide more contextual information to them.

Clustering is a data mining technique used in many fields such as machine learning, image processing or information retrieval. It is about splitting a given dataset into clusters (groups) such that the objects in a cluster are more similar to each other than objects in different clusters [6].

In this research, we apply various clustering methods on Europeana categorical data to automatically group objects based on specific facets such as title, format, creator, contributor etc. Categorical data are types of data which may be divided into groups e.g. gender, age group, educational level [7]. We propose a clustering method that is most appropriate to group Europeana Cultural Heritage objects (CHOs) by evaluating the clusters manually, on qualitative and quantitative level. We also aim at finding the best parametric setting for clustering on Europeana metadata and at finding the best way of clustering on Europeana metadata. We will perform three experiments to reach our purposes.

1.2. Research Questions

In this report, we try to answer the following research questions:

- How can we cluster the Europeana objects ?
- What is the clustering method with the best performance for clustering Europeana metadata ?
- What are the effects of different parameters on the cluster analysis results ?

This report is structured as follows: We explain the related work in section 2. We discuss our approach in section 3. We discuss first, second and third experiments in sections 4, 5 and 6 respectively. Section 7 includes the conclusion and discussion.

CHAPTER 2: RELATED WORK

Similar work has been done in [8] and [9]. The authors in [8] present a fast and scalable clustering algorithm and applied clustering to find semantic groups at different similarity levels for Europeana objects: A user could find culture heritage objects with five different levels of relatedness. Clustering process consists of three parts: 1) The objects are clustered on compression similarity. 2) Using genetic algorithms the important fields are automatically selected by taking an evolutionary approach to select the optimal solution based on a fitness function. 3) The records are hierarchically structured at different similarity levels.

The authors qualitatively evaluated intermediate results from UK records. They manually judged the cluster categories such as duplicate records, views of the same object, parts of an object, collections etc. per similarity level. The evaluation results have shown that clusters at higher

similarity levels are more accurate. The relevance of lower-level clusters is more difficult to evaluate. At higher levels, based on a single dimension of similarity they produce highly heterogeneous clusters (the heterogeneity is across clusters).

The authors in [9] show different techniques such as 'item similarity and 'typed similarity for cultural heritage data. The goal of these techniques is to help users to navigate and interpret the collections.

- *Item similarity*: Items which are similar to each other are collected. The PATHS project generated techniques to determine the similarity between items in cultural heritage collections by using Latent Dirichlet Allocation (LDA) to explore hidden 'topics' within the collection [9].

- *Typed similarity*: Various types of similarity can also be identified: Similar descriptions, similar author, similar people related, similar time period, similar events, similar locations. The similar pairs of items were mostly identified based on comparison of the text in the relevant fields of item's meta-data. E.g. The <dc:creator> field was used for identifying similar authors [9].

In the experiments we largely apply K-Means, PAM, EM, sIB, Farthest First, X-Means and Hierarchical clustering algorithms on Europeana objects while the authors in [8] used fast clustering based on compression similarity and they applied a genetic algorithm (to automatically select important fields) within Hierarchical clustering of CHOs at five similarity levels. So they applied a different way of hierarchical clustering. The manual evaluation is also performed to assess the clustering results; they chose 100 clusters at each level and asked 7 evaluators to categorise them. Each cluster is checked by at least two evaluators. The evaluators judged the clusters based on different categories per similarity level: same objects, views of the same object, parts of an object, derivative works, collections, thematic grouping, nonsense.

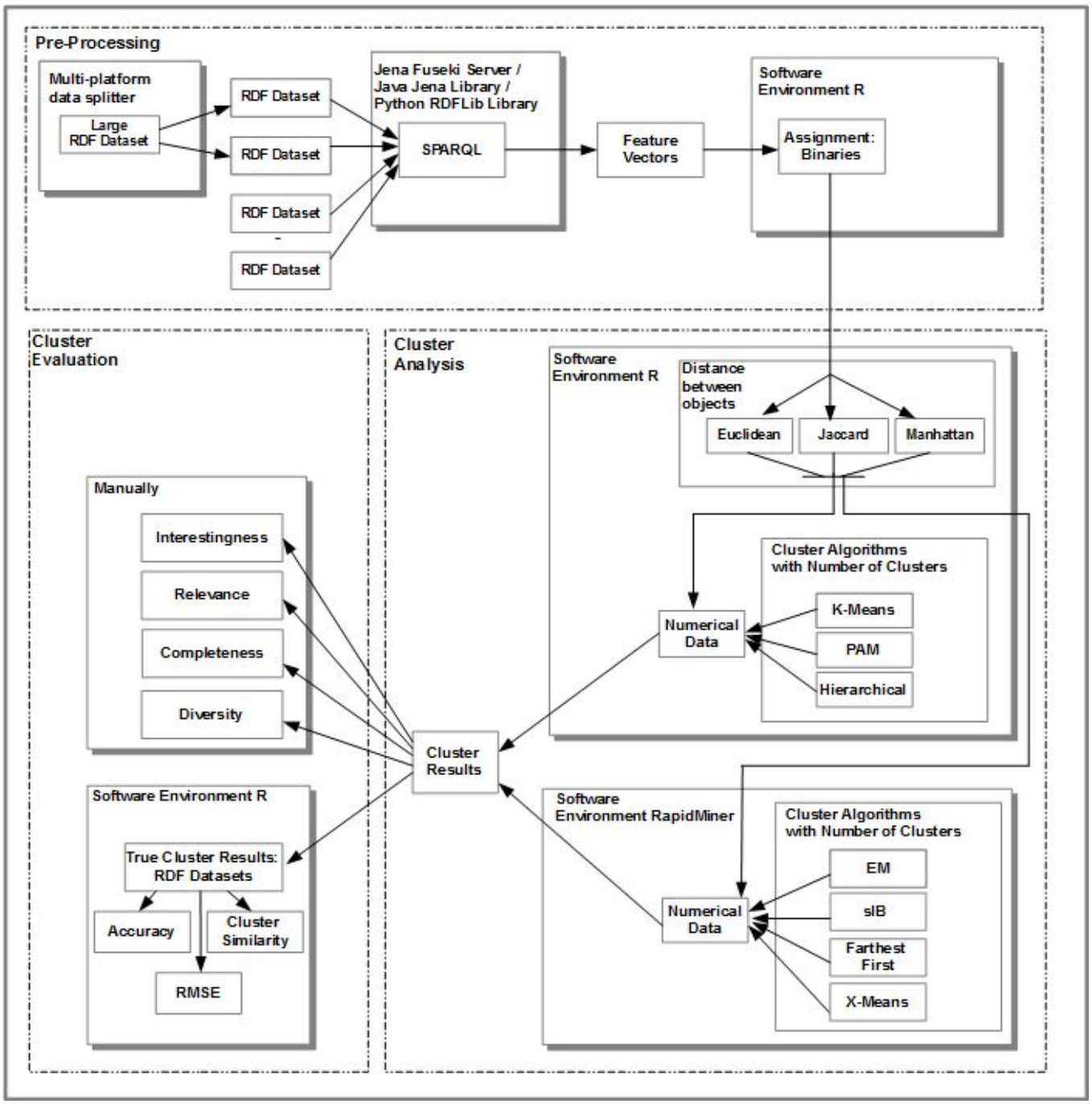
However, in our experiment we perform the manual evaluation differently; we ask two evaluators to assess all the clusters produced by Hierarchical on four factors: interestingness, relevance, completeness, diversity.

In [9] the similarity between all items in the three Europeana collections was computed in a pairwise form and the 25 items with the highest score are kept for each item. Similar pairs of items were identified using a range of Natural Language Processing techniques. However, in our experiments we use three similarity functions between the objects: Euclidean, Jaccard and Manhattan and apply the clustering methods we mentioned above.

CHAPTER 3: APPROACH

In this chapter we describe the approach as clustering process we used for the problem mentioned in Chapter 1. Figure 2 shows the clustering process.

Figure 2. Overview of the clustering process



We performed three experiments: In the first experiment we showed the way of clustering the Europeana objects. In the second experiment we applied the clustering algorithms by using a number of parameters to find the best parametric setting and the best clustering method. In the third experiment we used the best clustering method which we obtained from the second experiment to in order to see if it is good enough for Europeana. So Figure 2 also shows how we applied clustering in three experiments.

3.1. Pre-Processing

In the cluster analysis we used Europeana RDF datasets in N-Triple format. We obtained the data on [10]. Some RDF datasets have large size e.g. 281 MB. To deal with big Europeana datasets we can split data into sub-datasets with a multi-platform file splitter *HJSplit*. To extract metadata of RDF datasets on which we want apply the clustering algorithm we have used the RDF query language "SPARQL". We can process the SPARQL query on the platforms such as Jena Fuseki Server, Java with Jena Library, Python with RDFLib Library. By using Jena Library on Java we have read the datasets and extracted the metadata of RDF datasets. The output of this extraction process is the feature vectors. We extracted the values of following features: *title, type, format, date, publisher, relation, subject, identifier, source, edm type, creator, contributor, rights, provider, data provider, country* and *language.*

We assigned binaries to the field values: The output is the complete disjunctive table of a factor table. So if the value of a field is present for an object we assigned 1 to that field value otherwise we assigned 0. In assigning of binaries we excluded the fields with one or no value and missing values. Also, we did not use non-categorical fields such as identifier.

We generally used a similarity function to measure the closeness between the objects. This is also called the distance between the objects.

3.2. Cluster Analysis

After we computed the distances between the objects we got as output the numerical data. On the numerical data we applied the following clustering methods: *K-Means, PAM, Hierarchical, EM, sIB, Farthest-First* and *X-Means* with *k, k* is the number of clusters.

In the first and second experiments we selected *k* to be equal to the number of datasets since we assume that the RDF datasets are true clusters; we take the datasets as "ground truth". So in the evaluation of first and second experiments we want to compare the clusters produced by the algorithms to the datasets. In these experiments, as "gold standard" we selected the RDF datasets which have the provider "The European Library" (TEL). The reason for this is that TEL is currently enriching its collections descriptions, it provides collection-level data on subject, geographical and time coverage.

In the third experiment we selected different types of datasets, applied the best clustering method which we obtained from the second experiment and set up a threshold instead of selecting *k*.

3.2.1. Clustering Methods

In this section we will explain how the clustering methods used in the experiments work. We selected these methods since these algorithms are generally applicable in cluster tools we use: R, RapidMiner.

1) K-Means: Each cluster is associated with a centroid (center point) and each point is assigned to the cluster with the nearest mean and the mean is moved to the center of its cluster. This algorithm [11] consists of the following steps:

1. The dataset divided into k clusters and the data points are randomly assigned to the clusters.
2. For each data point:
   - 2a. Compute the distance from the data point to each cluster
   - 2b. If the data point is nearest to its own cluster, leave it where it is. If the data point is not nearest to its own cluster, move it into the nearest cluster.
3. Repeat the above step until a complete pass through all data points results in no data point moving from one cluster to another.

4. If the positions of the centroids did not change stop, else go to the Step 2.

One of the main reasons why we chose for K-Means because it is easier to understand and it is a fast algorithm.

2) Partition Around Medoids (PAM): This method [12] is very similar to K-Means since both are partitional algorithms, they both break datasets into groups and try to minimize the error. The only difference between these algorithms is that PAM works with medoids. The goal of PAM clustering method is to minimize the average dissimilarity of objects to their closest selected object. We chose for PAM since it produces groups that have the smallest within-group distances from each other and the furthest distances from the other groups.

3) Expectation-maximization (EM): This [13] is a popular probability density estimation method that is used in a variety of applications. It begins with an initial parameter estimation. The parameter values are used for computation of the likelihood. And it iterates until the clustering cannot be improved. In other words, until the clustering converges or the change is enough small. Each iteration includes two steps:

- *The expectation step* assigns the objects to clusters based on the current fuzzy clustering or parameters of probabilistic clusters.
- *The maximization step* finds the new clusters or parameters that maximize *SSE (Sum of squared errors)* in fuzzy clustering.

We selected EM because it is robust, easy and simple to implement.

4) Sequential Information Bottleneck (sIB): The sIB clustering algorithm [14] clusters co-occurence data such as text documents vs. words. So it clusters co-occurence data. This technique generates clusters where each document belongs to a single cluster. It tries to maximize the dependency between the clusters T and the words Y, computed by the mutual information $I(T, Y)$.

Let x be an initial assignment of documents to clusters t. This method sequentially draws a random document x from the clusters and finds a new cluster for it by minimizing a merging criterion. We chose for sIB since it is appropriate to use it on larger datasets.

5) Hierarchical: This algorithm groups in a hierarchy with a treelike structure according to the distance or similarity between them. It consists of the following steps [15]:

1. Assign each object to a cluster so that if you have N objects you have N clusters. Each cluster contains one object. Let the similarities (the distances) between the clusters be the same as the similarities between the objects they contain.
2. Find the most similar (closest) pair of clusters and merge them into one cluster so that you have one cluster less.
3. Calculate the similarities (the distances) between each of the old clusters and the new cluster.
4. Repeat steps 2 and 3 until all objects are clustered into a single cluster of size N.

Hierarchical detects sub-clusters within clusters and it does not require to specify the number of clusters beforehand. Also, it outputs a hierarchical tree which is more informative than the unstructured clusters produced by other algorithms. These are the main reasons why we selected Hierarchical clustering.

6) Farthest First: Farthest First cluster [16] is a variant of K-Means that places each cluster centre in turn at the point furthermost from the existing cluster centres. This point must lie within the data area. We chose for this method since it speeds up the clustering in most cases; less reassignment and adjustment is needed.

7) X-Means: The K-means algorithm [17] has three major shortcomings: It scales poorly computationally. The user has to provide the number of clusters and the search is inclined to local minima. The X-means algorithm resolves these shortcomings. It consists of the following two operations repeated until completion [17]:

X-Means:

1. Improve-Params
2. Improve-Structure

3. If $k > kmax$ stop and report the best scoring model found during the search.

The Improve-Params: It consists of running conventional K-means to convergence.

The Improve-Structure: It finds out if and where the new centroids should appear. This is obtained by letting some centroids split in two. There are two strategies for splitting:

Splitting idea 1: Pick one centroid, produce a new centroid nearby, run K-means to completion and see if the resulting model scores better. If it does, accept the new centroid. If it doesn't, return to the previous structure. Improve-Structure steps until X-means is complete.

Splitting idea 2: Try half the centroids. Choose half the centroids according to some heuristic criterion for how promising they are to split. Split them, run K-means, and see if the resulting model scores better. If so accept the split. Improve-Structure steps until X-means completes.

X-Means is an adjustment of K-Means which improves the speed and performance of clusters. This is why we selected X-Means as well.

3.2.2. Distance Measures

The three distances we used on the binary data in our cluster analysis are *Euclidean*, *Jaccard* and *Manhattan*. As you will see in Chapter 4, only in the first experiment we used no distance for some algorithms. This section includes the definitions of these distances:

1) Jaccard Distance: *A* and *B* are finite sample sets. Jaccard distance [18] between *A* and *B*,

dj(A, B) is computed as follows:

Jaccard coefficient, $J(A, B) = |A \cap B| / |A \cup B|$

$dj(A, B) = 1 - J(A, B) = (|A \cup B| - |A \cap B|) / |A \cup B|$

2) Euclidean Distance: The Euclidean distance [19] between a point *x (x1, x2,...etc.)* and a point *y (y1, y2,... etc)*:

$$d= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

3) Manhattan Distance: The manhattan distance [20] between the points is the sum of the differences of their corresponding components.

x (x1, x2,...etc.) and *y (y1, y2,... etc)* are two points. The manhattan distance between them:

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

3.3. Cluster Evaluation

In this section we describe the cluster evaluation process. In the evaluation of first and second experiments we used True Cluster Vectors. True Cluster Vectors are defined as follows: We assigned different numbers as cluster label to the objects based from which dataset they come; we take the datasets as "gold standard". And then we compared the clustering results to True Cluster Vectors to see whether the items are in the same way clustered as in the original datasets. So in the first and second experiments, to evaluate the quality of the clustering results we looked to the results of evaluation metrics based on True Cluster Vectors and to how the groups of objects are created by the clustering algorithm.

In the third experiment the evaluation is performed manually. We firstly provided the domain experts with the results of cluster method having the best performance and asked to them to judge the clustering results based on 4 factors.

3.3.1. Evaluation Metrics

For each clustering algorithm we mainly calculated the following evaluation metrics in the first and second experiments:

1) Accuracy: Fraction of correctly clustered vectors [21]. We computed accuracy as follows:

X: True Cluster Vectors

Y: Actual Cluster Vectors (so the cluster vectors obtained by the algorithm)

Accuracy: (Number of objects common to X_i and Y_i) / (Number of objects in *X*) where *i* is the row in vector

So with accuracy we want to find whether the items are clustered exactly in the same clusters as in the original datasets after running the clustering algorithms.

2) Cluster Similarity: We used cluster_similarity function [22] in *R* tool. *R* is a language and environment for statistical computing and graphics [23], it includes an effective data handling and storage facility, an integrated collection of intermediate tools for data analysis etc.
For two clusterings of the same dataset, this function calculates the similarity statistic specified of the clusterings from the comemberships of the clusterings from the comemberships of the observations. The comembership is defined as the pairs of observations that are clustered together.

3) Root Mean Square Error (RMSE): This metric [21] is used to measure the differences between values predicted by a model and the values actually observed from the environment.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2}{n}}$$ where X_{obs} is observed values and X_{model} is modelled values at time *i*.

4) Running Time: The execution time of each clustering algorithm we used.

We chose for accuracy, cluster similarity and RMSE since we want to make comparison between the clusters produced by the algorithm and Europeana RDF datasets with the provider "The European Library" as correct clusters. For the performance of algorithms it is also important to know how long does it take to execute the clustering algorithms.

In third experiment we defined a number of metrics for the manual evaluation. We made a cluster evaluation protocol which describes these metrics (See Section 3.3.1.1)

3.3.1.1. Cluster Evaluation Protocol

We made an evaluation protocol which describes in detail what the evaluators need to do for the manual evaluation of clustering results. We generally asked to the evaluators to judge the results based on 4 factors: interestingness, relevance, completeness and diversity. Below you will find the cluster evaluation protocol which we gave to the evaluators:

Manual Evaluation For Hierarchical Clustering

The evaluators will judge the the results of clustering based on the structure of clustering (hierarchical tree), on the metadata values of Europeana objects and on information about the datasets we used in this experiment (e.g. the aggregator of dataset, total number of objects per dataset etc.) given in the form of two spreadsheet tables. The evaluation is performed by looking to the the cluster parameters such as the number of clusters, structure of clustering: hierarchical tree and data size.

Besides this, we introduced the following terms to evaluate the clustering results manually: Interestingness, relevance, completeness and diversity. They are measured within the cluster. All these terms are subjective.

1) Interestingness: Interestingness of a cluster is related to the novelty, unexpectedness, excitingness, curiosity or usefulness to the user. For example, the cluster may include the object(s) which the user did not expect to see them together. If the cluster is interesting the evaluators label its interestingness as '1'. If not the evaluators label it as '0'. And also the evaluators indicate why the cluster is interesting or not interesting according to them.

2) Relevance: The relevance, also called compatibility is related to the similarity of objects with each other: A cluster is relevant if the objects in the cluster have the same metadata information. For example, the cluster is relevant because the objects have the same type, provider, data provider, country and language information.

If a cluster includes only two objects and each one has a different type, provider, data provider, country and language then this cluster is considered as 'irrelevant'. If the cluster is relevant the evaluators label its relevance as '1', otherwise they label it as '0'. And also the evaluators indicate why the cluster is relevant or irrelevant.

3) Completeness: We define the completeness of a cluster as the feeling that the cluster misses object(s). For example, a cluster can miss the object(s) while we think that the cluster should contain it /them. If the cluster is complete the evaluators label its completeness as '1' otherwise as '0'. And also they indicate why the cluster is complete or incomplete.

4) Diversity: It is related to the diversity of sub-clusters in a cluster. A cluster can be made of sub-clusters that are themselves quite different even if the cluster evaluated being makes sense. The evaluators measure the diversity of a cluster based on their pure guess and view. If the cluster is diverse the evaluators label its diversity as '1' otherwise as '0'. And also they indicate why the custer is diverse or not.

We assume that high interestingness, relevance and completeness will lead to high cluster performance while when the diversity decreases the cluster performance increases.

CHAPTER 4: EXPERIMENT 1

With this experiment we want to show how we can cluster Europeana objects. Performing Experiment 1 has lead to the setting of Experiment 2.

4.1. Experimental Setup

In this experiment we used three RDF datasets in N-Triple format:
92056_Ag_EU_TEL_a0022_Slovenia.nt, 92059_Ag_EU_TEL_a0138_Slovenia.nt, 92057_Ag_EU_TEL_a0156_Slovenia.nt.

All the objects in these datasets have the type either image or sound. Figure 3. shows an example of object from 92056_Ag_EU_TEL_a0022_Slovenia.rdf dataset.

Figure 3. Example of an object from 92056_Ag_EU_TEL_a00222.rdf dataset

We selected these datasets since they have the provider "The European Library"; we will use them as "gold standard" for the evaluation. All three datasets have in total 9413 objects.

Table 1. Number of Europeana Objects per dataset

Europeana RDF Dataset	**Number of Objects**
92056_Ag_EU_TEL_a0022_Slovenia	65
92059_Ag_EU_TEL_a0138_Slovenia	9230
92057_Ag_EU_TEL_a0156_Slovenia	118

Table 2. Number of different values per field

Number of different values per field					
Title	Type	Format	Date	Publisher	Relation
4333	4	0	319	832	14
Subject	Identifier	Source	Edmtype	Creator	Contributor
3512	9413	2	2	684	66
Rights	Provider	Data Provider	Country	Language	
0	1	1	1	1	

We extracted the metadata values for each dataset and combined all the metadata values into one table. We assigned binary values to these categorical values so that we can apply the methods on our data. In assigning of the binaries we did not use the fields such as *format, identifier, rights, provider, data provider, country* and *language;* these are either non-categorical variables or they have no or only one value.

We applied the following clustering techniques: K-Means, EM, Farthest First, Hierarchical and X-Means. In each clustering technique we selected for $k=3$ since we used three datasets as "ground truth"; as we earlier said, we used the collections of TEL since TEL is currently

enriching its collections descriptions. So it gave us appropriate set of collections and metadata about them, for items that already in Europeana. For the datasets which do not have the provider "The European Library" the number of clusters is not known, we need to find a way to determine the optimum number of clusters or use a clustering technique in which we do not need to specify the number of clusters.

In K-Means, EM and Farthest First algorithms we used no distance. So only in Hierarchical and X-Means we used Euclidean distance.

To evaluate the clustering results we generated True Cluster Vectors: We assigned the integers 1, 2 and 3 to the objects from 92056_Ag_EU_TEL_a0022_Slovenia.nt, 92059_Ag_EU_TEL_a0138_Slovenia.nt, 92057_Ag_EU_TEL_a0156_Slovenia.nt datasets respectively.

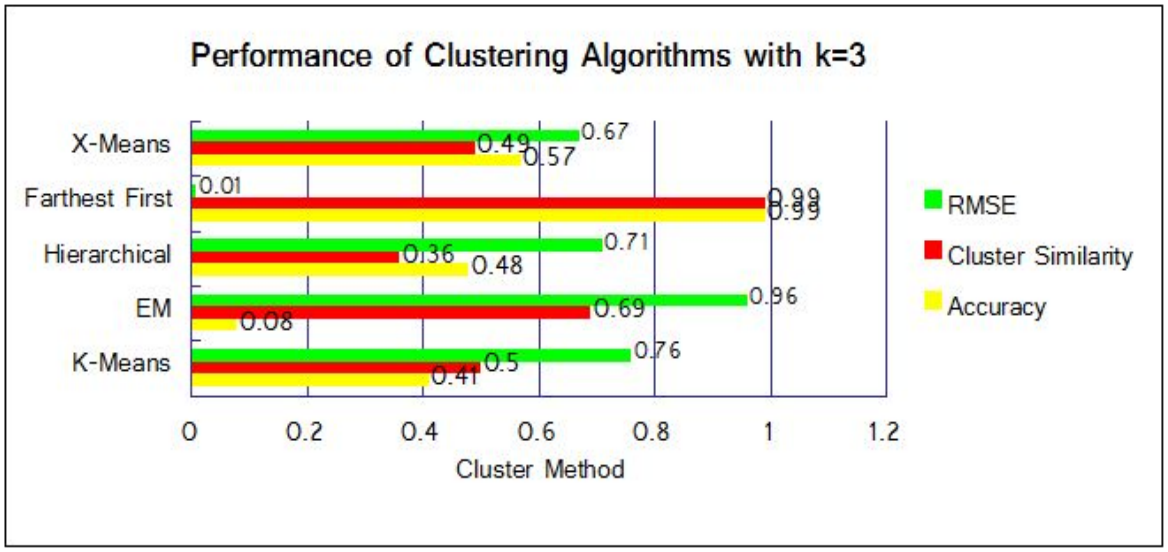
Table 3. Number of items per cluster produced by several clustering algorithms

Clustering Algorithm	**Number of Objects**		
	Cluster 1	**Cluster 2**	**Cluster 3**
K-Means	65	3993	5355
EM	898	892	7623
Hierarchical	2024	4415	2974
Farthest First	65	9230	118
X-Means	3967	5379	67
True Cluster	65	9230	118

4.2. Cluster Evaluation Results

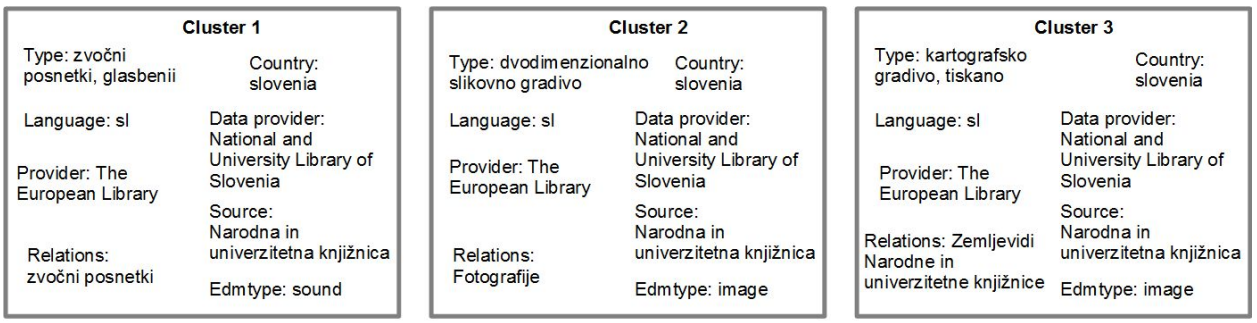
Figure 4 shows the accuracy, cluster similarity and RMSE values of each algorithm. We recall that $k=3$ and in K-Means, EM, Farthest First no distance has been used.

Figure 4. Results of Evaluation Metrics for 5 clustering algorithms for 3 Europeana datasets



Only the items in the clusters generated by Farthest First algorithm are correctly clustered (See Figure 4). In Figure 5 we can see the feature information of the clusters. E.g. All items in the clusters have the country 'slovenia'. So this figure also shows the correct clusters.

Figure 5. Feature information of items per cluster generated by FarthestFirst Method



On the results of evaluation metrics we see that only Farthest First had the highest accuracy and cluster similarity among all the methods. As we said, only the items of clusters produced by this algorithm are correctly clustered. EM performed worse than all other methods in accuracy while Hierarchical performed worse in cluster similarity.

With this experiment we presented a way to cluster Europeana objects. This process of clustering may be useful for Europeana. We will later compare the evaluation results of first and second experiments with each other.

We will describe Experiment 2 in Chapter 5.

CHAPTER 5: EXPERIMENT 2

With this experiment we want to find the "best" setting and the "best" clustering algorithm. For doing this, in this experiment we used a number of *parameters* for each clustering technique:

- Transformation method for categorical data: Binaries
- Selection of fields in the metadata
- Distance measure: Euclidean, Jaccard, Manhattan
- *k*: {2,3,4,5,6} where *k* is the number of clusters

As a transformation method for categorical data, we automatically generated a table of binaries; we assigned binary values to each attribute value per object: If the attribute value is present we assigned 1 otherwise 0.

In this experiment we used the same pre-processing and clustering methods as in the previous experiment. But here we used two more methods (sIB and PAM), the parameters mentioned above and another datasets with smaller sizes.

5.1. Experimental Setup

Per *k* we used in total six RDF datasets in N-Triple format:

92056_Ag_EU_TEL_a0022_Slovenia.nt, 92057_Ag_EU_TEL_a0156_Slovenia.nt, 92058_Ag_EU_TEL_a0246_Slovenia.nt, 92051_Ag_EU_TEL_a0310_Slovenia.nt, 92052_Ag_EU_TEL_a0311_Slovenia.nt, 92053_Ag_EU_TEL_a0312_Slovenia.nt.

We chose for six RDF datasets since they have the provider "The European Library" so that we can use them as "gold standard". Also, they do not have very large size.

For each *k* we applied the same transformation method such as in the previous experiment: We extracted all the metadata values, combined them and assigned binaries to the categorical

values. We did not assign binaries to the values of fields such as format, identifier, source, rights, provider, data provider, country and language; these are either non-categorical or they have no or only one value. And also for each *k* we performed the clustering algorithms with Euclidean, Jaccard and Manhattan distances. In this experiment we applied seven techniques: K-Means, PAM, EM, sIB, Hierarchical, Farthest First, X-Means.

To evaluate the quality of clustering results we generated True Cluster Vectors such as in the previous experiment: Based on *k* we assigned the integers between 1 and 6 to the objects in datasets respectively. For example, if $k=2$ and we have 2 datasets then the objects in one dataset take 1 and the objects in second dataset take 2.

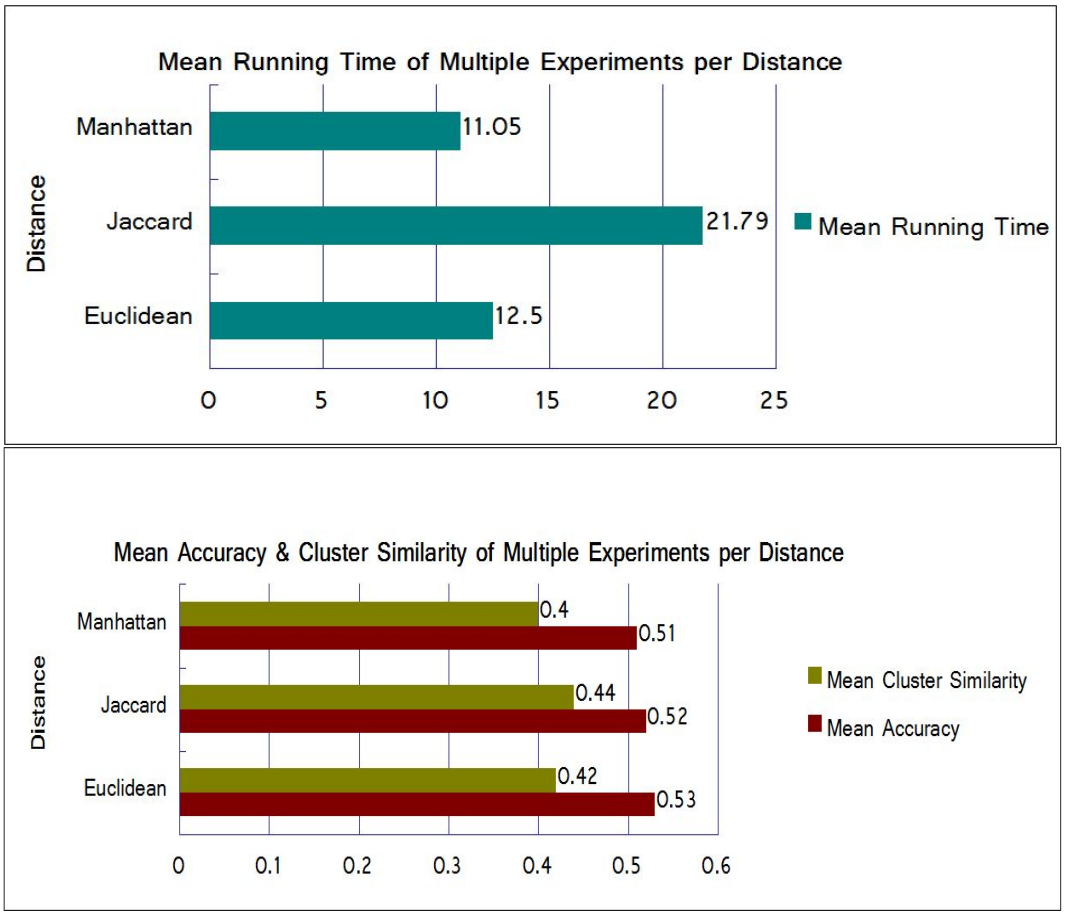
5.2. Cluster Evaluation Results

This chapter includes the evaluation results of Experiment 2 based on distance measures, number of clusters, mean evaluation metrics, performance of methods.

5.2.1. Distance Measures

As we see in Figure 6 we computed the mean values of accuracy, cluster similarity and running time of all three distances by taking into account the clustering results of each algorithm per number of clusters and per distance, so all 15 cases. Euclidean and Jaccard perform better than Manhattan in cluster similarity and accuracy.

Figure 6. Mean Evaluation Metrics of Multiple Experiments per Distance



5.2.2. Number of Clusters

When we look only to the number of clusters we see that most algorithms have the highest accuracy, cluster similarity and the lowest running time, RMSE when k is 2 (See Table 5 in Appendix A). So in all the distance measures we get the algorithms with the best performance when k = 2. Also, in the cluster results of all the distance measures we notice that the accuracy, cluster similarity of most algorithms decrease and their running time increases when k increases.

5.2.3. Mean Evaluation Metrics

We computed the mean running time, cluster similarity and also accuracy of all the algorithms by taking into account all the distances (See Figures 7 and 8). Hierarchical algorithm has the highest accuracy, cluster similarity and the lowest execution time. It is interesting to see that also EM has the highest accuracy. PAM and sIB have the lowest cluster similarity while K-Means is the least accurate. sIB has the highest mean execution time among the algorithms. When we used Jaccard and Euclidean in sIB we got the lowest cluster similarity. However, we got the highest cluster similarity when we used Jaccard and Euclidean distances in Hierarchical and Farthest First (See Figures on Appendix A).

Figure 7. Mean Running Time per Clustering Algorithm

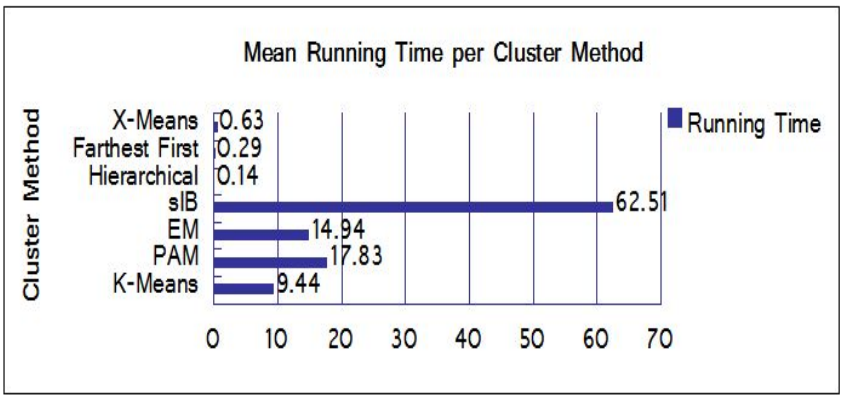
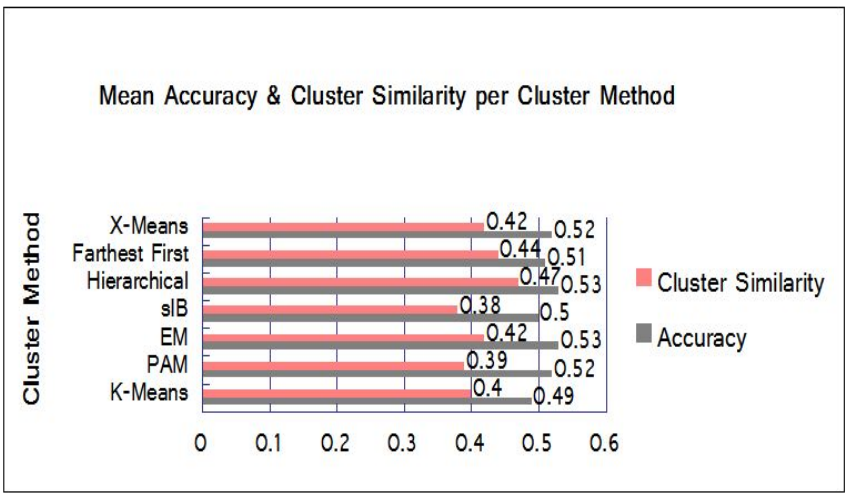


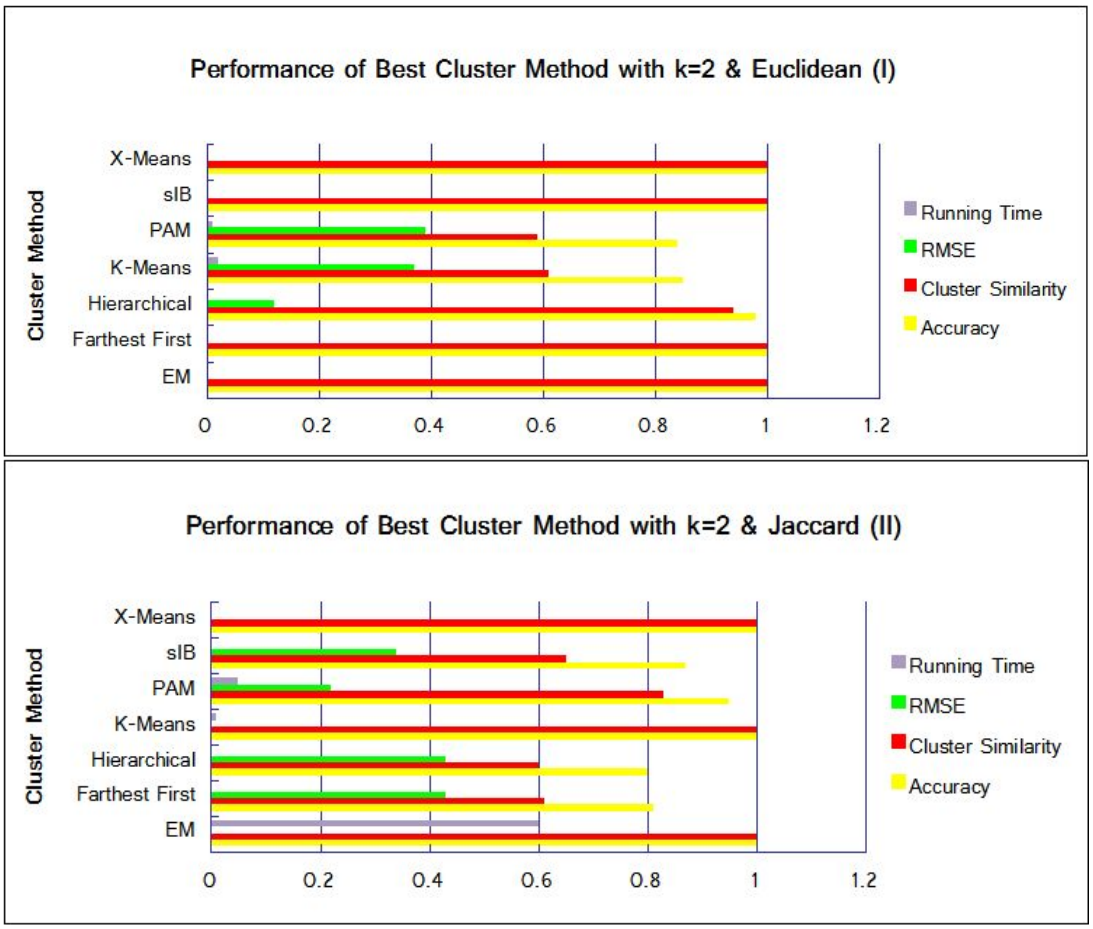
Figure 8. Mean Accuracy and Cluster Similarity per Clustering Algorithm

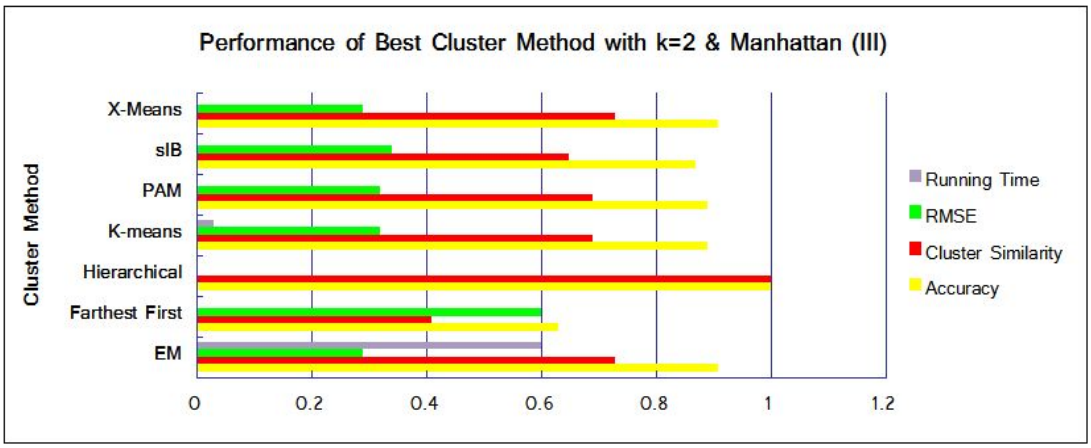


5.2.4. Performance of Best Clustering Algorithms

We looked to the clustering results of each algorithm per number of clusters and per distance, so to all the cases. Figure 9 shows the performances of best methods based on True Cluster Vectors of Experiment 2. So they have the highest accuracy, cluster similarity and the lowest execution time. This also means that most objects in the clusters generated by these algorithms are correctly clustered. The objects that have the same field value (e.g. type, publisher, relation and source) are grouped together.

Figure 9. Performance of Best Clustering Algorithms





1) K-Means

K-Means is easy to understand and implement and it may generate tighter clusters than e.g. Hierarchical. However, in the experiment K-Means algorithm was slower in all distances than Hierarchical, Farthest First and X-Means. As we have seen in the experiment its accuracy and cluster similarity were worse than these three algorithms. Also, K-Means does not work well with clusters of different size, it cannot handle outliers and noisy data.

2) PAM

PAM is an expensive algorithm; it finds the medoids because it compares each medoid with the whole dataset at each iteration. In the experiment PAM was slower than the most algorithms (even slower than K-Means). However, this method is more robust than K-Means since it minimizes a sum of dissimilarities instead of a sum of squared Euclidean distances. We also have seen in the experiment that PAM performed better in accuracy than some algorithms such as K-Means or Farthest First. Its cluster similarity was also higher than that of sIB.

3) EM

As we earlier said EM and Hierarchical had the highest accuracy and EM is better in the cluster similarity than sIB and PAM. EM also maximizes the likelihood. However, EM can take large amount of time to create the clusters. This increases the cost. It took longer time to run EM than K-Means, Hierarchical, Farthest First and X-Means.

4) sIB

The performance of sIB was worst among all the methods. It took the largest amount of time to run this algorithm. It also has the lowest cluster similarity and accuracy on average. This algorithm is not so popular as K-Means or Hierarchical.

5) Farthest First

Farthest First is a fast and greedy algorithm. As we noticed in the running time results of this experiment, Farthest First was one of the fastest algorithms. It is similar to K-Means and very suitable for large amounts of data. It determines each remaining center by choosing the point farthest from the set of already selected centers. It had also a high accuracy and cluster similarity on average.

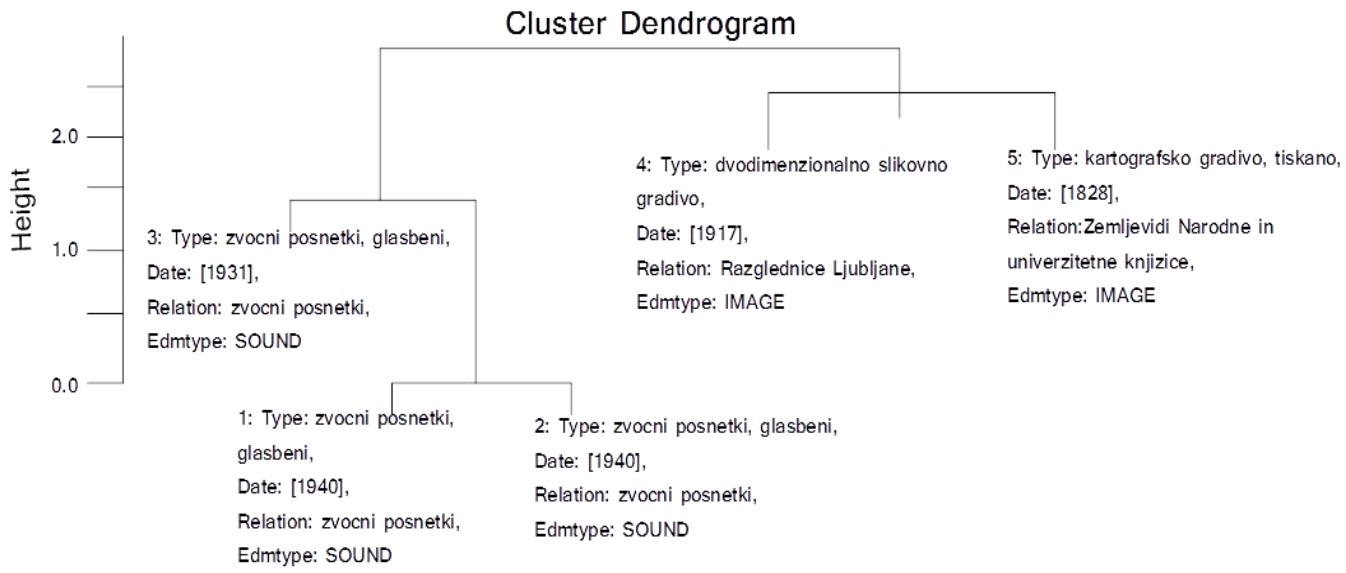
6) X-Means

Although X-Means is developed to quickly estimate the number of clusters since after each run of K-Means it makes local decisions by splitting some centroids it performed faster and better than most algorithms in this experiment.

7) Hierarchical

Hierarchical algorithm computes the complete hierarchy of the clusters. It is only effective at splitting small amount of data. However, it has a logical structure. It also gives a good visualization about how different parts of data changes from other data. If we consider all the results of evaluation metrics it performed much better than all other methods; it is the fastest and most accurate algorithm with the highest cluster similarity score on average.

Figure 10. Example of Hierarchical Clustering - Dendrogram



All the algorithms performed better when $k=2$. But we would not select for $k=2$ for an experiment of e.g. 100 datasets. As we have seen in the results, the overall performance of the Hierarchical method is the best one among the seven methods.

When we compare the results of first and second experiments with each other where we take $k=3$, based on "gold standard" we concluded that Farthest First had the best performance in first experiment. But the best method was Hierarchical in second experiment. This is due to the different size of datasets or different features used in first experiment.

We learned from second experiment that it is difficult to define the best parametric setting and best clustering method only based on a number of experiments. So this does not completely match our expectations.

CHAPTER 6: EXPERIMENT 3

Clustering is generally used to detect new structures in the data instead of reproducing known structure. So if a clustering method discovers structures that are different from their datasets, this could be seen in fact as a good and admirable result [24]. For this reason, we want to

see whether the clustering algorithm has found desirable clusters. Also, since we do not know the true clustering structure of our data we need the knowledge of domain experts.

In this experiment we evaluate the clustering results manually with the help of domain experts at Europeana. So instead of taking datasets as 'ground truth' our evaluation is based on a number of subjective terms. The manual evaluation is about the results of clustering method with the best performance which we obtained from Experiment 2, namely Hierarchical. With this experiment we want to know if it is good enough. We used the same pre-processing method as in Experiment 2. In this chapter we describe in detail how we performed the cluster analysis and how the domain experts at Europeana judged the clustering results manually. Finally we discuss the evaluation results.

6.1. Experimental Setup

In this section we describe how we selected RDF datasets for this experiment. Two domain experiments at Europeana provided us with a list of RDF datasets which are interesting; these were the datasets having the aggregators 'Landesarchiv Baden-Württemberg', 'BHL Europe' and 'Centre Virtuel de la Connaissance sur l'Europe'. To evaluate the clustering results in a correct way it is necessary that all the objects in the datasets on which we apply clustering exist. For all datasets from three aggregators as mentioned above we checked automatically using Sparql on Java if all Europeana records in the dataset can be found on Europeana.eu (if they exist). So we have written a program that opens the dataset file, checks the first Europeana.eu URLs and indicates whether URLs are dead or not (See Appendix C). We selected only the datasets in which all the records are online available. So even if one object in the dataset did not exist we did not select that dataset.

We provided the evaluators with the results of one experimental setting including table of overview experimental setup as seen below, table of objects with related clusters, datasets and metadata values, hierarchical cluster tree, table of datasets with related number of total objects, number of text objects, number of image objects, aggregator, aggregated country etc. You can find more information in Cluster Evaluation Protocol in Section 3.3.1.1.

Accordingly, we made a selection with some elements such as the number of datasets, clustering method, dissimilarity matrix or threshold as follows:

We randomly choose three datasets from each of three aggregators. As usual, after extraction of metadata values we excluded the fields such as format, creator, rights and identifier because they are either non-categorical, they have one or no value. So we used in total 9 datasets. We set the threshold to 3.5. The algorithm produced 7 clusters.

In Chapter 5 we claimed that hierarchical performed better than all other algorithms we used in Experiment 2. In this experiment we want to know if it is good enough for Europeana. We also argued in Chapter 5 that the dissimilarity metrics such as Euclidean performed better than Jaccard and Manhattan in accuracy. For this reason, we selected Euclidean distance in this experiment.

Table 4: Overview experimental setup for the manual evalution

Clustering Method	Hierarchical with type: Agglomerative and linkage criteria: Complete linkage clustering
Number of Europeana datasets used	9
Number of total Europeana objects	2018
Number of used attributes	13
Attributes that are not used	Format, Creator, Rights, Identifier
Dissimilarity matrix between objects	Euclidean
Threshold	3.5

6.2. Cluster Evaluation Results

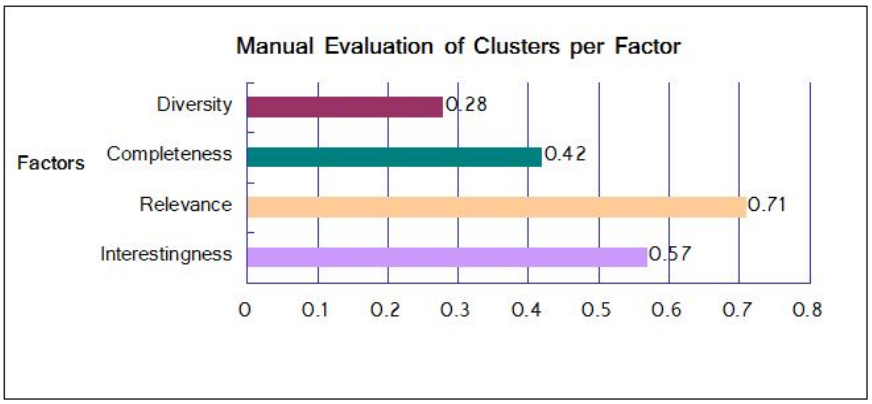
This chapter includes the evaluation results of Experiment 3 based on factors as mentioned in Chapter 3: interestingness, relevance, completeness, diversity. Besides, we also discuss the structure of clustering, data size, threshold or number of clusters. The evaluators provided us with the results of manual evaluation (See Appendix D).

Measuring the interestingness is a difficult task because it is quite based on the personal feelings, opinions. For example, we see in the results that the evaluators labeled one of the clusters as interesting even though almost all objects are from the same dataset. This is a situation which we should not expect to see. We notice that only 3 out of 7 clusters are labeled as 'not interesting' and only 2 clusters are labeled as 'not relevant'.

Cluster 4 includes only one object while this object should belong to Cluster 3. It is not clear why Cluster 4 has only one object. This cluster decreased the overall completeness, interestingness, relevance and diversity of clusters. We see that most clusters include the sub-clusters which are not different from each other. Only two clusters are labeled as 'diverse'. Figure 11 shows the proportions of interesting, relevant, complete and divers clusters related to the number of clusters.

Also, we did not compute the inter-annotater agreement between the evaluators since each of the evaluators judged different clusters.

Figure 11: Manual Evaluation Scores of Clusters per Factor for Hierarchical Clustering



We think that it is difficult to know if the hierarchical clustering is good enough for Europeana only based on the results of manual evaluation of one experiment since the evaluation is subjective, so the assessment is based on the knowledge and view of domain experts. Another

problem is that they judge the clustering result itself and we cannot compare their assessment to the true clustering because it is not known. However, the results of manual evaluation of hierarchical method show that except of completeness (0.42) we generally have good scores for the evaluation factors. We have a low score for the diversity (0.28) which is good for the cluster performance as we expected.

6.2.1. Number of clusters

The algorithm did not require to specify the maximal number of iterations or the number of clusters in advance as input because the clusters are obtained from cutting the tree at different levels. We may specify the number or clusters in advance or we can to set a threshold and stop clustering when the distance between the clustering above the threshold, so we can set the termination criteria to stop when the distance between nearest clusters exceeds a predefined threshold [25]. By setting the threshold to 3.5 the algorithm the algorithm produced 7 clusters while the objects come from 9 datasets. This is because of the fact that the algorithm has put more datasets in one cluster and the objects from the same dataset are in more clusters. For example, the objects from the datasets DS1, DS2, DS3 are in cluster 1 and the objects from the dataset DS7 are in cluster 3 and 4. So the clusters can be overlapping (they can have the datasets in common). In the hierarchical algorithm one cluster can be completely contained within another cluster, no other type of overlap between the clusters is allowed.
All objects must have the distance smaller than 3.5 in order to consider them a cluster. We still do not know the correct number of clusters. And also the algorithm is less suitable to the outliers [26].

6.2.2. Structure of clustering and data size

In this method the clustering is used to find a meaningful structure in data. As it computes and shows the complete hierarchy it may be helpful to see the whole clustering of data. But as we have seen in the experiment at a large dataset the algorithm produced a large hierarchical tree. As a result of this, it becomes difficult to interpret the results on hierarchical tree. For example, the large hierarchical tree (dendrogram) made difficult to measure the diversity of a cluster. This problem is related to the presentation of results, so it is not a problem within the clustering method itself.

CHAPTER 7: CONCLUSION AND DISCUSSION

This report presents the results of experiments with respect to the cluster analysis we performed using Europeana RDF data. In this report we applied different clustering methods on Europeana RDF datasets: K-Means, PAM, EM, sIB, Hierarchical, Farthest First and X-Means.

We performed three experiments with different goals: With Experiment 1 we wanted to show how we can cluster Europeana objects. In Experiment 1 we used True Cluster Vectors, Euclidean distance only in Hierarchical and X-Means algorithms and we took $k=3$ where k is the number of clusters. With Experiment 2 we wanted to find the best setting to cluster Europeana items and the best clustering method by using some parameters and True Cluster Vectors. So in both experiments we used RDF datasets with the provider "The European Library" as "gold standard". With Experiment 3 we wanted to know if the hierarchical clustering is good enough for Europeana by performing manual evaluation.

In the first experiment we presented a way to cluster Europeana objects. In second experiment we defined the following cluster parameters which have effect on the quality of categorical clustering: Transformation method for categorical data, selection of fields in the metadata, distance measures and number of clusters. So the quality of clustering is certainly dependent on these parameters as well.

As we earlier said, the features that are used in the cluster analysis (selection of fields) is one of these parameters. In the experiments we noticed that we get different clustering results when we take different datasets as input for categorical clustering. This is because of the fact that Europeana datasets may include different type of information, so different attributes or values. For example, assume that we have two datasets. The objects in one dataset may have no format while another dataset may include format. The clustering methods will produce two different clusters for two datasets when we use each one of them apart for categorical clustering. Therefore we think that it is difficult to select which fields are most appropriate for clustering. We used only one transformation method for extracted categorical data: binaries.

If we compare the evaluation results of Experiment 1 and Experiment 2 with each other where we take $k=3$, based on True Cluster Vectors we concluded that Farthest First was the best method for categorical clustering while the best method is Hierarchical in Experiment 2. This is due to the different size of datasets e.g. larger or different features used in Experiment 1.

The results in Experiment 2 showed that the algorithms were more accurate and they had the higher cluster similarity when we used Euclidean and Jaccard distances. Also, all the algorithms showed better performance when $k=2$. However, we would not select $k=2$ for an experiment of e.g. 100 datasets.

Hierarchical has the best performance in the average accuracy, cluster similarity and running time (based on the results of evaluation metrics) compared to other techniques. In this method we can cluster Europeana objects in the structure of a binary tree that combines similar groups of objects at each step. It was obvious in Experiment 2 that sIB was not one of good techniques for categorical clustering in which we use binaries.

Experiment 3 showed that the clusters produced by the hierarchical algorithm generally have high interestingness, relevance and low diversity. Except of the completeness the clustering results are good for the cluster performance. However, the problem with manual evaluation by the domain experts is that the evaluation is subjective and the evalutors judge the clustering result itself. We cannot compare their assessment to the 'ideal' clustering since it is not known. We also think that we still do not know the correct number of clusters in the hierarchical clustering as well. However, since the clusters are obtained from cutting the tree at the similarity level (threshold) we can easily set a threshold when we know which similarity level is best to cut the tree.

We conclude that it is difficult to define the best parametric setting and best clustering method only based on a number of experiments. It is also hard to know if the hierarchical clustering is good enough for Europeana based on the manual evaluation we performed. So we did not entirely achieve our goals with two experiments. However, we have shown a way to cluster Europeana objects which may be useful for Europeana.

Future work might include an adjustment of hierarchical or other methods to determine the optimum number of clusters, to perform other experiments with big Europeana datasets to define the best clustering and setting.

BIBLIOGRAPHY

1. Europeana - Facts and Figures.

URL http://pro.europeana.eu:9580/documents/900548/1338813/Latest+factsheet.pdf

2. G. Macgregor. Collection-level descriptions: metadata of the future? Library Review, Vol 52 Iss: 6, pp.247 – 250, 2003.

URL http://eprints.rclis.org/6007/1/Macgregor_CLD_OAversion.pdf

3. K.M. Wickett, A. Isaac, K. Fenlon, M. Doerr, C. Meghin, C. L. Palmer, J. Jett. Modeling Cultural Collections for Digital Aggregation and Exchange Environments. *CIRSS Technical Report 201310-1, University of Illinois at Urbana-Champaign*, 2013.

URL https://www.ideals.illinois.edu/handle/2142/45860

4. V. Charles. Europeana Data Model (EDM) and EDM for libraries, *Days of Knowledge Organization*, 2012.

URL

http://www.jbi.hio.no/bibin/korg_dagene/korg2012/europeana_data_model__charles_korg2012.pdf

5. Metadata, *Australian National Data Service*, 2014.

URL http://ands.org.au/guides/metadata-awareness.pdf

6. M. Halkidi, Y. Batistakis, M. Vazirgiannis. On Clustering Validation Techniques, *Intelligent Information Systems Journal, Kluwer Publishers, 17(2-3): 107-145*, 2001.

URL http://web.itu.edu.tr/sgunduz/courses/verimaden/paper/validity_survey.pdf

7. Categorical Data - Introduction to Statistics, 1997.

URL http://www.stat.yale.edu/Courses/1997-98/101/catdat.htm

8. S. Wang, A. Isaac, V. Charles, R. Koopman, A. Agoropoulou, T. van der Werf. Hierarchical structuring of Cultural Heritage objects within large aggregations. *OCLC Research, Europeana Foundation*, 2013.
URL http://arxiv.org/pdf/1306.2866v1.pdf

9. M. Stevenson, A. Otegi, E. Agirre, N. Aletras, P. Clough, S. Fernando, A. Soroa. Semantic Enrichment of Cultural Heritage Content in PATHS. *Personalised Access to Cultural Heritage Spaces (PATHS)*, 2013.
URL
http://www.paths-project.eu/eng/Resources/Semantic-Enrichment-of-Cultural-Heritage-content-in-PATHS

10. Europeana RDF Datasets
URL http://data.europeana.eu/download/2.0/datasets/nt/

11. K-Means Clustering Overview, *Improved Outcomes Software.*
URL
http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/K-Means_Clustering_Overview.htm

12. S. Horvath. Weighted Network Analysis Applications in Genomics and Systems Biology - Chapter 8: Clustering Procedures and Module Detection, *University of California*, 2011.
URL
http://books.google.nl/books?id=ZCh06NgMFesC&pg=PA181&lpg=PA181&dq=Partitioning+Around+Medoids+algorithm+edu&source=bl&ots=UeCNLfhlGo&sig=OTTy3_bp5eBG_nRh_sKEw-wZr9M&hl=nl&sa=X&ei=PVpmU7HLG4L-PMTOgKgG&ved=0CE4Q6AEwBTgU#v=onepage&q=Partitioning%20Around%20Medoids%20algorithm%20edu&f=false

13. J. Han, M. Kamber, J. Pei. Data Mining Concepts and Techniques - Chapter 11.1: Probabilistic Model-Based Clustering, *The Morgan Kaufmann Series in Data Management Systems,* 2012.
URL
http://books.google.nl/books?id=pQws07tdpjoC&pg=PA505&dq=expectation+maximization+cluster&hl=nl&sa=X&ei=SK5mU8qZHYzrOfHlgDg&ved=0CF0Q6AEwBA#v=onepage&q=expectation%20maximization%20cluster&f=false

14. J. Peltonen, J. Sinkkonen, S. Kaski. Sequential Information Bottleneck for Finite Data, *Helsinki University of Technology and University of Helsinki*, 2004.
URL http://www.machinelearning.org/proceedings/icml2004/papers/158.pdf

15. M. Matteucci. A Tutorial on Clustering Algorithms - Hierarchical Clustering Algorithms, *Politecnico Di Milano*.
URL http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html

16. N. Sharma, A. Bajpai, R.Litoriya. Comparison of the various clustering algorithms of weka tools, *Jaypee University of Engineering and Technology*, 2012.
URL http://www.ijetae.com/files/Volume2Issue5/IJETAE_0512_13.pdf

17. D. Pelleg, A. Moore. X-Means: Extending K-means with Efficient Estimation of the Number of Clusters, *Carnegie Mellon University*, 2007.
URL http://www.cs.cmu.edu/~dpelleg/download/xmeans.pdf

18. J.M. Philips. Lecture 4 - Jaccard Similarity and Shingling. *University of Utah,* 2013.
URL http://www.cs.utah.edu/~jeffp/teaching/cs5955/L4-Jaccard+Shingle.pdf

19. Euclidean and Euclidean Squared. *Improved Outcomes Software.*
URL
http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Euclidean_and_Euclidean_Squared_Distance_Metrics.htm

20. Manhattan. *Improved Outcomes Software.*
URL
http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Manhattan_Distance_Metric.htm

21. K. Hammouda. A Comparative Study of Data Clustering Techniques. *University of Waterloo*, 2004.
URL http://www.pami.uwaterloo.ca/pub/hammouda/sde625-paper.pdf

22. J. A. Ramey. Evaluation of Clustering Algorithms. *The Comprehensive R Archive Network*, 2013.
URL http://cran.r-project.org/web/packages/clusteval/clusteval.pdf

23. The R Project For Statistical Computing
URL http://www.r-project.org/

24. I. Färber, S. Günnemann, H. P. Kriegel, P. Kröger, E. Müller, E. Schubert, T. Seidi, A. Zimek. On Using Class-Labels in Evaluation of Clusterings, *RWTH Aachen University, Ludwig-Maximilians-Universität München*, 2010.
URL http://eecs.oregonstate.edu/research/multiclust/Evaluation-4.pdf

25. M. Soss. Cluster Analysis - Hierarchical Clustering Algorithms, *McGill University*, 1999.
URL http://cgm.cs.mcgill.ca/~soss/cs644/projects/siourbas/sect5.html#hfigure4

26. J. Gao. Clustering Lecture 3: Hierarchical Methods, *University at Buffalo,* 2012.
URL http://www.cse.buffalo.edu/~jing/cse601/fa12/materials/clustering_hierarchical.pdf

APPENDIX A: CLUSTER EVALUATION RESULTS - EXPERIMENT 2

Figure 12. Mean Running Time of Clustering Algorithms per Distance - Experiment 2

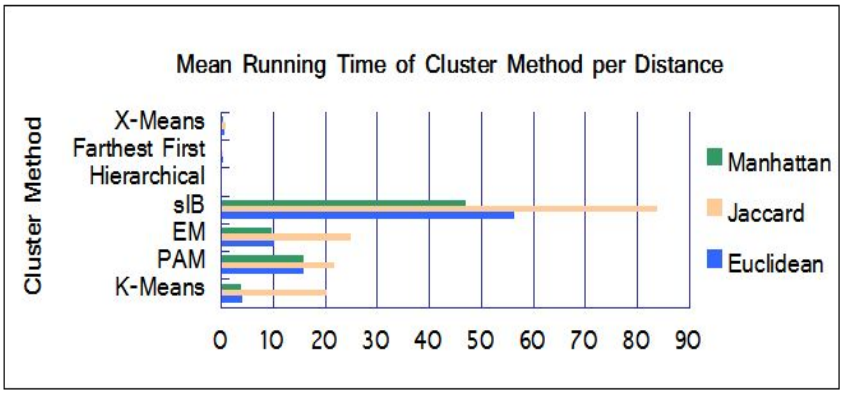


Figure 13. Mean Accuracy of of Clustering Algorithms per Distance - Experiment 2

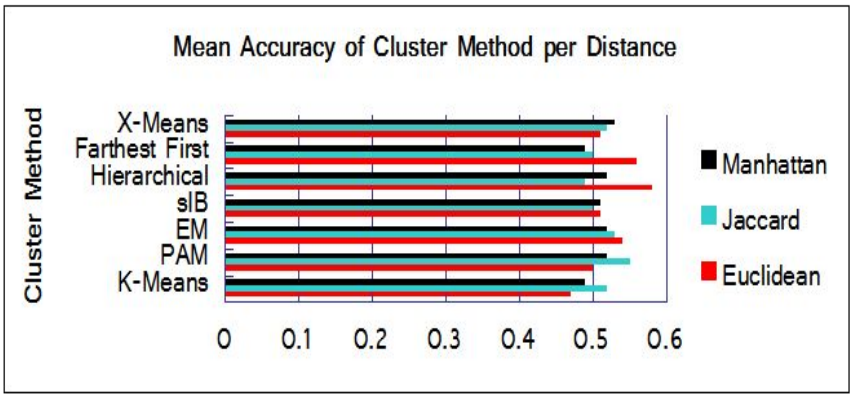


Figure 14. Mean Cluster Similarity of of Clustering Algorithms per Distance - Experiment 2

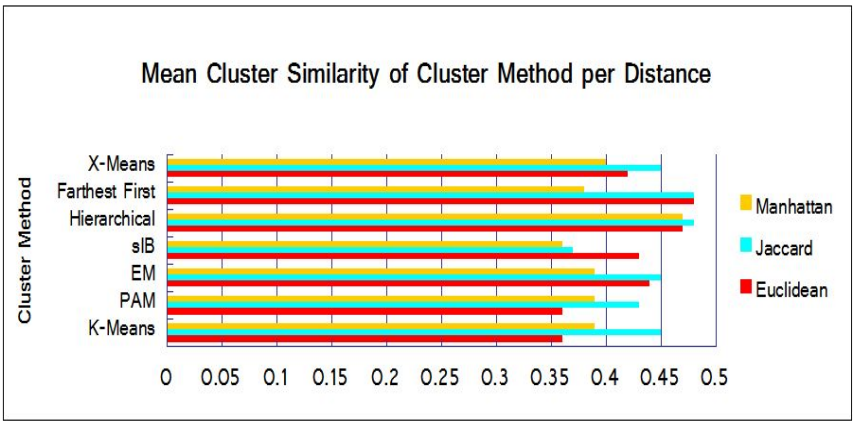


Table 5. Cluster Evaluation Results - Experiment 2

Assignment: Binaries, Distance: Euclidean & k =2					Assignment: Binaries, Distance: Euclidean & k =3				
Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time	Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time
K-Means	0.85	0.61	0.37	0.02	K-Means	0.76	0.54	0.48	0.02
PAM	0.84	0.59	0.39	0.01	PAM	0.75	0.52	0.49	0.02
EM	1	1	0	0	EM	0.77	0.55	0.47	0.8
sIB	1	1	0	0	sIB	0.68	0.46	0.56	0.1
Hierarchical	0.98	0.94	0.12	0	Hierarchical	0.86	0.66	0.36	0
Farthest First	1	1	0	0	Farthest First	0.8	0.55	0.44	0
X-Means	1	1	0	0	X-Means	0.74	0.44	0.5	0

Assignment: Binaries, Distance: Euclidean & k =4					Assignment: Binaries, Distance: Euclidean & k =5				
Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time	Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time
K-Means	0.28	0.25	1.6	1.51	K-Means	0.17	0.21	1.86	6.47
PAM	0.28	0.25	1.59	1.08	PAM	0.31	0.21	1.43	7.58
EM	0.28	0.25	1.59	2.01	EM	0.31	0.21	1.42	17.28
sIB	0.33	0.25	1.66	37.24	sIB	0.33	0.2	1.53	71.81
Hierarchical	0.24	0.26	1.59	0.23	Hierarchical	0.37	0.23	1.47	0.17
Farthest First	0.2	0.35	1.36	0.3	Farthest First	0.41	0.3	1.07	0.5
X-Means	0.28	0.25	1.59	0.35	X-Means	0.27	0.2	1.59	1.24

Assignment: Binaries, Distance: Euclidean & k =6				
Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time
K-Means	0.32	0.23	1.37	12.28
PAM	0.36	0.23	1.33	70.87
EM	0.36	0.23	1.31	30.59
sIB	0.24	0.26	1.87	173.0
Hierarchical	0.48	0.28	1.09	0.23
Farthest First	0.42	0.27	1.06	0.7
X-Means	0.28	0.21	1.49	1.13

Assignment: Binaries, Distance: Jaccard & k =2				
Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time
K-Means	1	1	0	0.01
PAM	0.95	0.83	0.22	0.05
EM	1	1	0	0.6
sIB	0.87	0.65	0.34	0
Hierarchical	0.8	0.6	0.43	0
Farthest First	0.81	0.61	0.43	0
X-Means	1	1	0	0

Assignment: Binaries, Distance: Jaccard & k =3				
Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time
K-Means	0.58	0.45	0.64	0.02
PAM	0.77	0.57	0.47	0.03
EM	0.73	0.55	0.51	0.4
sIB	0.71	0.54	0.53	0.1
Hierarchical	0.27	0.69	0.84	0
Farthest First	0.19	0.71	0.89	0
X-Means	0.54	0.45	0.67	0

Assignment: Binaries, Distance: Jaccard & k =4				
Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time
K-Means	0.43	0.34	1.17	12.68
PAM	0.37	0.3	1.3	16.13
EM	0.3	0.26	1.54	13.23
sIB	0.33	0.25	1.61	32.56
Hierarchical	0.71	0.54	0.74	0.27
Farthest First	0.64	0.48	0.69	0.3
X-Means	0.43	0.34	1.17	0.39

Assignment: Binaries, Distance: Jaccard & k =5					Assignment: Binaries, Distance: Jaccard & k =6				
Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time	Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time
K-Means	0.37	0.254	1.33	35.82	K-Means	0.25	0.23	1.64	53.52
PAM	0.36	0.245	1.11	21.80	PAM	0.34	0.25	1.08	71.07
EM	0.32	0.217	1.37	25.21	EM	0.34	0.23	1.26	85.14
sIB	0.3	0.2	1.47	205.12	sIB	0.3	0.21	1.38	182.0
Hierarchical	0.3	0.3	0.97	0.35	Hierarchical	0.4	0.27	0.99	0.41
Farthest First	0.52	0.35	0.68	0.5	Farthest First	0.34	0.27	1.15	0.8
X-Means	0.34	0.25	1.06	1.09	X-Means	0.33	0.23	1.11	3.35

Assignment: Binaries, Distance: Manhattan & k =2					Assignment: Binaries, Distance: Manhattan & k =3				
Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time	Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time
K-Means	0.89	0.69	0.32	0.03	K-Means	0.77	0.57	0.47	0.17
PAM	0.89	0.69	0.32	0	PAM	0.79	0.58	0.45	0.01
EM	0.91	0.73	0.29	0.6	EM	0.77	0.57	0.47	0.7
sIB	0.87	0.65	0.34	0	sIB	0.7	0.52	0.54	0.1
Hierarchical	1	1	0	0	Hierarchical	0.83	0.64	0.4	0
Farthest First	0.63	0.41	0.6	0	Farthest First	0.82	0.59	0.41	0
X-Means	0.91	0.73	0.29	0	X-Means	0.78	0.58	0.46	0

Assignment: Binaries, Distance: Manhattan & k =4				
Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time
K-Means	0.28	0.25	1.59	4.09
PAM	0.29	0.25	1.59	6.36
EM	0.28	0.25	1.59	7.12
sIB	0.33	0.25	1.66	27.55
Hierarchical	0.23	0.27	1.65	0.16
Farthest First	0.2	0.36	1.35	0.3
X-Means	0.28	0.25	1.59	0.27

Assignment: Binaries, Distance: Manhattan & k =5				
Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time
K-Means	0.29	0.21	1.73	7.89
PAM	0.31	0.21	1.42	6.82
EM	0.31	0.21	1.41	15.48
sIB	0.26	0.2	1.75	58.13
Hierarchical	0.24	0.24	1.36	0.2
Farthest First	0.41	0.3	1.07	0.4
X-Means	0.31	0.21	1.4	0.53

Assignment: Binaries, Distance: Manhattan & k =6				
Cluster Algorithm	Accuracy	Cluster Similarity	RMSE	Running Time
K-Means	0.23	0.23	1.5	7.09
PAM	0.36	0.23	1.31	65.74
EM	0.37	0.23	1.3	25.04
sIB	0.29	0.21	1.48	150.0
Hierarchical	0.31	0.22	1.58	0.25
Farthest First	0.42	0.27	1.06	0.6
X-Means	0.37	0.23	1.3	1.20

Table 6. Number of different values per attribute - Pre-processing - Experiment 2

	Number of different values				
Attribute	**k=2**	**k=3**	**k=4**	**k=5**	**k=6**
Title	173	195	1603	1738	2017
Type	3	4	6	6	6
Format	0	0	0	0	0
Date	95	104	113	115	175
Publisher	64	74	75	76	100
Relation	12	13	96	110	122
Subject	68	87	88	88	165
Identifier	183	205	1753	1910	2199
Source	1	22	22	22	22
Edmtype	2	2	3	3	3
Creator	81	81	154	199	199
Contributor	66	66	100	129	129
Rights	0	0	0	0	0
Provider	1	1	1	1	1
Data Provider	1	1	1	1	1
Country	1	1	1	1	1
Language	1	1	1	1	1

APPENDIX B: PRE-PROCESSING: EXTRACTION FEATURES - EXPERIMENT 1

SparqlRDF.java (Java Jena Library)

```
package sparql;
import com.hp.hpl.jena.query.Query;
import com.hp.hpl.jena.query.QueryExecution;
import com.hp.hpl.jena.query.QueryExecutionFactory;
import com.hp.hpl.jena.query.QueryFactory;
import com.hp.hpl.jena.query.ResultSet;
import com.hp.hpl.jena.query.ResultSetFormatter;
import com.hp.hpl.jena.rdf.model.*;
import com.hp.hpl.jena.util.FileManager;
import java.io.*;
public class SparqlRdf extends Object {
    public static void main(String[] args) throws FileNotFoundException {
        String inputFileName  = "D:/92059_Ag_EU_TEL_a0138_Slovenia_2.nt";
        // create an empty model
        Model model = ModelFactory.createDefaultModel();
        InputStream in = FileManager.get().open(inputFileName);
       if (in == null) {
            throw new IllegalArgumentException( "File: " + inputFileName + " not found");
        }
        // read the RDF/XML file
        model.read(in, null, "N-TRIPLE");
        String queryString =
                "PREFIX edm: <http://www.europeana.eu/schemas/edm/> " +
                "PREFIX ore: <http://www.openarchives.org/ore/terms/> " +
                "PREFIX dc: <http://purl.org/dc/elements/1.1/> " +
                "PREFIX dct: <http://purl.org/dc/terms/> " +
                "SELECT DISTINCT ?proxyProvider (GROUP_CONCAT(DISTINCT ?title)
AS ?titles) (GROUP_CONCAT(DISTINCT ?type) AS ?types) "
```

```
                + "(GROUP_CONCAT(DISTINCT ?format) AS ?formats)
(GROUP_CONCAT(DISTINCT ?date) AS ?dates) "
                + "(GROUP_CONCAT(DISTINCT ?publisher) AS ?publishers)
(GROUP_CONCAT(DISTINCT ?relation) AS ?relations) "
                + "(GROUP_CONCAT(DISTINCT ?subject) AS ?subjects)
(GROUP_CONCAT(DISTINCT ?identifier) AS ?identifiers) "
                + "(GROUP_CONCAT(DISTINCT ?source) AS ?sources)
(GROUP_CONCAT(DISTINCT ?edmtype) AS ?edmtypes) "
                + "(GROUP_CONCAT(DISTINCT ?creator) AS ?creators)
(GROUP_CONCAT(DISTINCT ?contributor) AS ?contributors) "
                + "(GROUP_CONCAT(DISTINCT ?right) AS ?rights)
(GROUP_CONCAT(DISTINCT ?provider) AS ?providers) "
                + "(GROUP_CONCAT(DISTINCT ?dataProvider) AS ?dataProviders)
(GROUP_CONCAT(DISTINCT ?country) AS ?countrys) "
                + "(GROUP_CONCAT(DISTINCT ?language) AS ?languages)\n"
                + "WHERE {\n" +
                        "?proxyProvider ore:proxyIn ?aggregationProvider .\n" +
                            "OPTIONAL {?proxyProvider dc:title ?title }\n" +
                            "OPTIONAL {?proxyProvider dc:type ?type }\n" +
                            "OPTIONAL {?proxyProvider dc:extent ?format }\n" +
                            "OPTIONAL {?proxyProvider dc:date ?date }\n" +
                            "OPTIONAL {?proxyProvider dc:publisher ?publisher }\n" +
                            "OPTIONAL {?proxyProvider dc:relation ?relation }\n" +
                            "OPTIONAL {?proxyProvider dc:subject ?subject }\n" +
                            "OPTIONAL {?proxyProvider dc:identifier ?identifier }\n" +
                            "OPTIONAL {?proxyProvider dc:source ?source }\n" +
                            "OPTIONAL {?proxyProvider edm:type ?edmtype }\n" +
                            "OPTIONAL {?proxyProvider dc:creator ?creator }\n" +
                            "OPTIONAL {?proxyProvider dc:contributor ?contributor }\n"
                            +
                            "OPTIONAL {?aggregationProvider dc:rights ?right }\n" +
```

```
                                "OPTIONAL {?aggregationProvider
edm:provider ?provider }\n" +
                                "OPTIONAL {?aggregationProvider
edm:dataProvider ?dataProvider }\n" +
                              "?aggregationEuropeana
ore:aggregates ?aggregationProvider .\n" +
                                "OPTIONAL {?aggregationEuropeana
edm:country ?country }\n" +
                                "OPTIONAL {?aggregationEuropeana
edm:language ?language }\n" +
              "} GROUP BY ?proxyProvider\n" ;
        Query query = QueryFactory.create(queryString);
        QueryExecution qe = QueryExecutionFactory.create(query, model);
        ResultSet results = qe.execSelect();
        PrintStream out = new PrintStream(new
    FileOutputStream("D:/92059_Ag_EU_TEL_a0138_Slovenia_2.csv"));
        System.setOut(out);
        ResultSetFormatter.outputAsCSV(out, results);
        qe.close();
    }
}
```

APPENDIX C: PRE-PROCESSING: CHECKING EXISTENCE OF EUROPEANA OBJECTS - EXPERIMENT 3

CheckEuropeanaRecords.java (Java Jena Library)

```
package sparql;

import com.hp.hpl.jena.query.Query;
import com.hp.hpl.jena.query.QueryExecution;
import com.hp.hpl.jena.query.QueryExecutionFactory;
import com.hp.hpl.jena.query.QueryFactory;
import com.hp.hpl.jena.query.QuerySolution;
import com.hp.hpl.jena.query.ResultSet;
import com.hp.hpl.jena.query.ResultSetFormatter;
import com.hp.hpl.jena.rdf.model.*;
import com.hp.hpl.jena.util.FileManager;
import java.net.*;
import java.io.*;
import java.util.Iterator;
import java.util.List;
import java.util.Properties;

public class SparqlRdf extends Object {
    public static void main(String[] args) throws FileNotFoundException,
UnsupportedEncodingException {
        String inputFileName =
"F:/EUROPEANA_RDF_DATASETS/CHECK_RDF_DATASETS/BHL/08703_Ag_EU_NBN.
rdf";
        // create an empty model
        Model model = ModelFactory.createDefaultModel();
        InputStream in = FileManager.get().open(inputFileName);
```

```
    If (in == null) {
        throw new IllegalArgumentException( "File: " + inputFileName + " not found");
    }
    // read the RDF/XML file
    model.read(in, null, "RDF/XML");
    String queryString =
            "PREFIX edm: <http://www.europeana.eu/schemas/edm/> " +
            "PREFIX ore: <http://www.openarchives.org/ore/terms/> " +
            "PREFIX dc: <http://purl.org/dc/elements/1.1/> " +
            "PREFIX dct: <http://purl.org/dc/terms/> " +
            "SELECT DISTINCT ?recordHtml \n"
            + "WHERE {\n" +
                    "?proxyProvider ore:proxyIn ?aggregationProvider .\n" +
                    "?aggregationEuropeana edm:landingPage ?recordHtml .\n" +
        "}\n" ;

    Query query = QueryFactory.create(queryString);
    QueryExecution queryexec = QueryExecutionFactory.create(query, model);
    ResultSet setofresults = queryexec.execSelect();
    while(setofresults.hasNext()){
        QuerySolution qsolution = setofresults.next();
        String outcome = qsolution.toString();
        System.out.println("Europeana URL:  " +
outcome.split("<")[1].split(">")[0]);
        checkurls(outcome.split("<")[1].split(">")[0]);
        System.out.println("\n");
    }
}
```

```
    //Check whether Europeana.eu URL exists
    public static void checkurls(String url){
            try {
              Properties systemSettings = System.getProperties();
              systemSettings.put("proxySet", "true");
              systemSettings.put("http.proxyHost","proxy.mycompany.local") ;
              systemSettings.put("http.proxyPort", "80") ;

              URL u = new URL(url);
              HttpURLConnection con = (HttpURLConnection) u.openConnection();

              sun.misc.BASE64Encoder encoder = new sun.misc.BASE64Encoder();
              String encodedUserPwd =
                  encoder.encode("domain\\username:password".getBytes());
              con.setRequestProperty
                  ("Proxy-Authorization", "Basic " + encodedUserPwd);
              con.setRequestMethod("HEAD");
              System.out.println
                  (con.getResponseCode() + " : " + con.getResponseMessage());
              if(con.getResponseCode() == HttpURLConnection.HTTP_OK) {
                   System.out.println("Record exists!");
              } else {
                   System.out.println("Record does not exist!");
              }
            }
            catch (Exception e) {
              e.printStackTrace();
            }
          }
}
```

APPENDIX D: RESULTS OF MANUAL EVALUATION - EXPERIMENT 3

Table 7: Results of manual evaluation provided by two domain experts

Clusters	Interestingness	Why?	Relevance	Why?	Completeness	Why?	Diversity	Why?
Cluster 1	0	The grouping doesn't make much sense.	0	Besides their provenance (country and provider) these objects are very different.	1	At least in the selected datasets, there are no other items like these ones.	1	The three datasets in this cluster have not much in common except their provenance.
Cluster 2	0	It merges object from 3 datasets on the EU. They all come from the same institution, and are quite trivial.	1	They are all connected to the EU.	1	At least in the selected datasets, there are no other items like these ones.	1	There are clearly three different sub-clusters - but the metadata doesn't show it so it's difficult to extract them.
Cluster 3	1	Ok, even though it's almost all objects from DS7.	1	The objects are quite similar.	0	It misses at least the object of cluster 4.	0	The cluster seems quite homogeneous.
Cluster 4	0	A cluster with one object is not interesting.	0	Not measurable - there is just one object.	0	The object is similar to the ones of cluster 3.	0	There is just one object.

Cluster 5	1	Ok even if all the objects belong to the same dataset.	1	Metadata for the objects are quite similar: publisher, type, relation, provider, country.	0	The cluster 5 and 6 should be together.	0	The cluster is homogeneous: same type of objects, same topics.
Cluster 6	1	Ok even if all the objects belong to the same dataset.	1	Metadata for the objects are quite similar: relation, data provider, type.	0	The cluster 5 and 6 should be together.	0	The cluster is homogeneous: same type of objects, same topics.
Cluster 7	1	Ok the objects belong to the same more abstract objects (different volumes of the same journal.	1	Metadata are nearly the same for all the objects.	1	Seems complete.	0	Same type of resource.