TIN2009-11005
DAMASK

Data-Mining Algorithms with Semantic Knowledge

PROYECTO DE INVESTIGACIÓN
PROGRAMA NACIONAL DE INVESTIGACIÓN FUNDAMENTAL,
PLAN NACIONAL DE I+D+i 2008-2011
ÁREA TEMÁTICA DE GESTIÓN: Tecnologías informáticas

Deliverable D5
Adapting K-means for including numerical, categorical and semantic multi-valued attributes

Authored by
Ferran Mata
Aïda Valls
Sergio Martínez Lluís

Document information

project name:	DAMASK	
Project reference:	TIN2009-11005	
type of document:	Deliverable	
file name:	D5.pdf	
version:	1.0	
authored by:	F. Mata, A. Valls, S. Martínez	19/07/2012
co-authored by		
released by:	A. Valls	25/07/2012
approved by:	Co-ordinator	Antonio Moreno

Document history

version	date	reason of modification
1.0	19.July.2012	Adaptation of *k*-means for including numerical, categorical and semantic multi-valued attributes.

Table of Contents

1 Introduction and motivation

This document corresponds to Task 2 of the DAMASK project. Task T2 is focused on the goal O2 of the project: design of a clustering method based on ontologies. The inputs of this task are: (1) a data matrix object × attribute (e.g. touristic destinations) and (2) a domain ontology. Based on those inputs, a method for automatically building clusters is needed. During the clustering, contextual knowledge provided by the domain ontology is used. Finally, an automatic interpretation process of the clusters is required, in order to obtain a semantic description of the clusters that can help the user in his/her decision making tasks.

This deliverable is the result of the subtask T2-4 developed from month 21 until month 26. The complete schedule of the tasks in the DAMASK project is given in Figure 1. Task 2-4 corresponds to the study of the adaptation of the traditional clustering algorithms to permit the use of semantic similarity measures based on ontologies and linguistic terms.

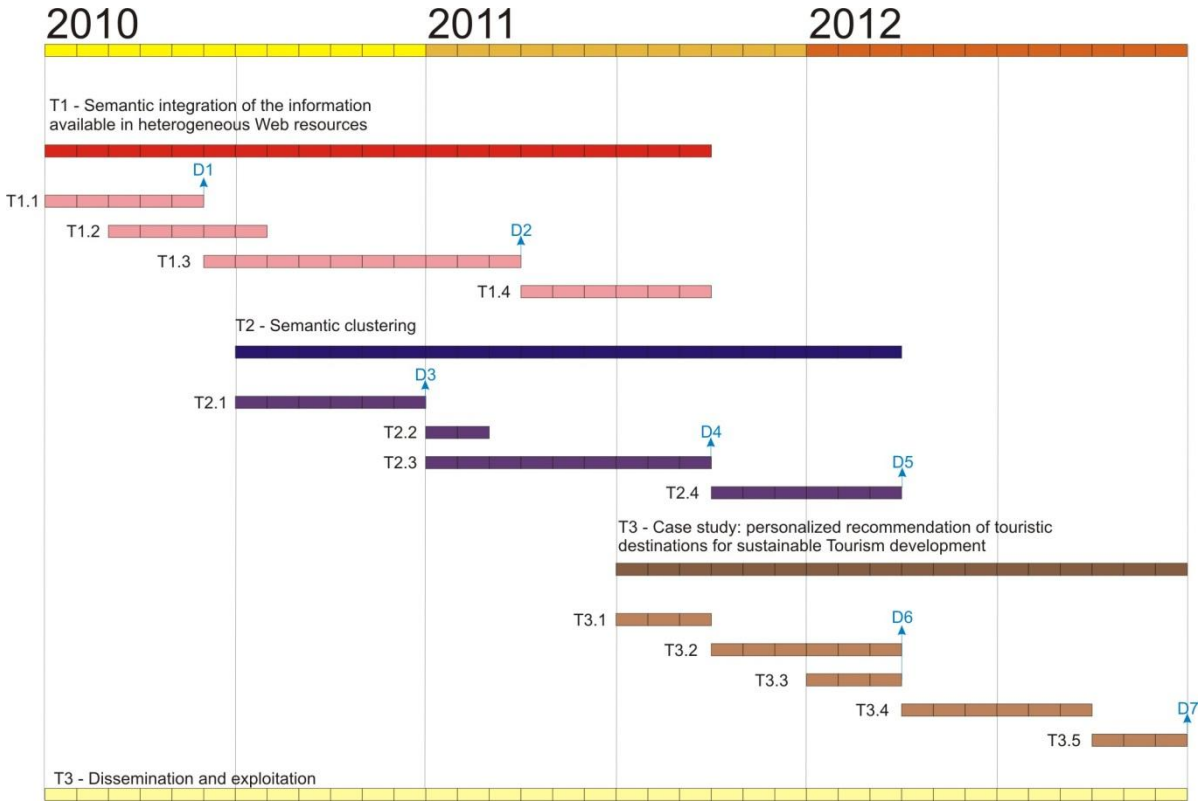


Figure 1: Tasks of DAMASK

An extensive analysis of the weak points of the existing semantic similarity measures was made in Task 2.1 and presented in deliverable D3. In deliverable D4 is presented a new semantic similarity measure for pairs of objects that solve the limitations of the previous approaches with respect to the improvement of the clustering of objects explaining how to include this semantic similarity measures into some clustering algorithms.

In Deliverable D2 a state of the art of clustering methods is presented. Finally, the *k*-means method is selected according to its properties: high scalability and simplicity. The *k*-means method was initially proposed for numerical data (Forgy, 1965; MacQueen, 1967). Later extensions considered its applicability to categorical data. In this document we will extend the k-means algorithm in order to deal also with semantic attributes, those with a semantic interpretation by means of the use of an ontology. We assume that all the values of a given semantic attribute are represented by a concept in the ontology. Different ontologies could be used for each attribute. In the system developed in the DAMASK project, all the attributes use the same domain ontology: the Tourism Ontology developed by the experts that participate in the project (see DAMASK report 3.1).

The work presented in this deliverable has been done mainly as part of the Master Thesis of Ferran Mata having as advisor Dr. Aïda Valls (member of DAMASK project). Sergio Martínez has also worked in this Ph.D Thesis in the topic of centroid construction for semantic variables.

The document is organized as follows. First, we present classic *k*-means algorithm, discussing the operators used for numerical and for categorical attributes. The case of semantic attributes is presented in section 2, proposing new operators. In particular, we focus on the design of a new method for generating a centroid (i.e. prototype) for a group of objects described by a semantic multi-valued attribute. Afterwards, in section 3 the new version of the k-means for introducing semantic attributes is presented. The implementation and testing are discussed in section 4.

1.1 Families of clustering algorithms

Partitioning a set of objects into homogeneous clusters is fundamental. The operation is required in a number of data analysis tasks, such unsupervised classification or segmentation of large heterogeneous data sets into smaller homogeneous subsets that can be easily managed, separately modelled and analysed. Clustering is a knowledge discovery technique used to gather a set of objects in groups according to their similarity. There are two main types of clustering approaches in function of the properties of the generated clusters: hierarchical and partitioning clustering.

- **Partitional clustering**: The aim of this type of clustering is to create a division of the set of objects into *k* groups, where *k* is a pre-specified number that indicates the amount of desired clusters ($k \leq N$). These partitions do not overlap with each other. Hence, each data object belongs to only one of the *k* subsets.
- **Hierarchical clustering:** Constructs a taxonomical structure of the set of objects, creating a hierarchical decomposition of the given data set, and producing a binary tree known as a *dendogram*. The *root* node represents the whole data set, and each *leaf* node is a single object; the rest of intermediate nodes correspond to clusters that group similar objects. Overlapping between clusters is also not allowed.

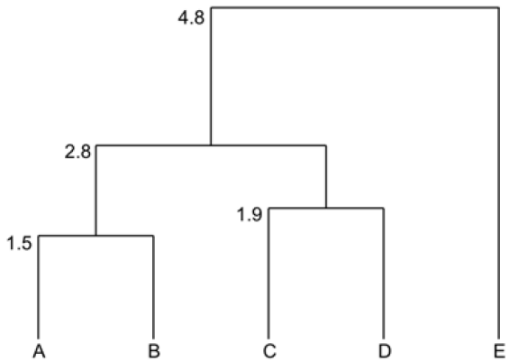


Figure 2: Dendogram

1.2 Partitional clustering

In partitional clustering, a set of *N* objects are assigned to *k* clusters. Each cluster must have at least one object, and each object must belong to just one cluster. It is important to remark that the number of clusters (*k*) is predefined by the user. It is usually done on the basis of some specific criterion, so one of the important factors in partitional clustering is the criterion function (Hansen et al., 1997).

Partitioning methods are divided into two major subcategories depending on which type of representation the clusters have:

- Centroid: These algorithms represent each cluster by using some sort of centre of gravity of the objects, with an artificially created prototype. This approach has the problem of defining a method for generating this prototype. The method to obtain the centroid is usually some sort of average of the values of the objects. If the objects are just numerical values, the Euclidean average is a perfect centroid. But if the objects are non-numerical, finding an averaging function is not trivial. It is even more difficult when the objects have various attributes of different types. Different approaches have been defined using dissimilarity measures for categorical objects, such as Huang(Huang, 1998) and Gupta (Gupta et al., 1999).

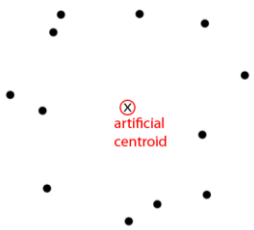


Figure 3: Artificial centroid representation. High precision.

- Medoid: The aim of these algorithms is to use one of the cluster objects to represent the cluster. The selected object is the one that its average dissimilarity to all the objects in the cluster is minimal, i.e. it is the most centrally located point in the cluster. This approach avoids the problem of calculating an artificial prototype. It only requires the definition of a distance between objects. The cost of using the medoid method instead of the more complex centroid method is the precision of the representation. For instance, a cluster with all of its objects at

more or less the same distance will not have a very representative medoid (except for the case where their distance is 0).

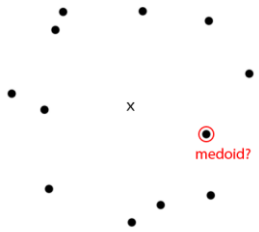


Figure 4: Medoid representation. Low precision.

The most important algorithm for partitional clustering is called *k*-means. Several variations of this algorithm can be found in the literature. Some of them are reviewed in the next section.

1.3 The classic *k*-means algorithm

k-means is the most well-known centroid algorithm (Forgy, 1965; MacQueen, 1967). *k*-means aims to partition *N* objects into *k* clusters. Each object belongs to the cluster with its nearest centroid, which is the cluster's representative. *k* is predefined number.

The *k*-means clustering gives a formal definition as an optimization problem: find the cluster centers and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized (MacQueen, 1967).

The problem is NP-hard, and that means that only have approximate solutions. The most common *k*-means algorithm only finds a local optimum.

These are the steps of the *k*-means clustering algorithm:

Determine the number of desired *k* partitions
Repeat until there are no changes in the centroids {
    Start **selecting *k* initial centroids randomly**
    Compute the **distance** of each object to the *k* centroids.
    Assign each object to the cluster where its centroid has the lowest distance.
    Compute a new centroid for the computed clusters.
}

The *k*-means has some advantages and disadvantages that are numbered below:

- Advantages:
    - The algorithm is simple and, despite it is an NP-hard problem, it is also fast; what makes it appropriate to cluster large data sets.
    - It tends to converge in just a small number of iterations, what makes this algorithm very efficient.
- Disadvantages:

- The iterative procedure of *k*-means cannot guarantee convergence to a global optimum. This leads to some problems:
  - It is sensitive to the selection of the initial partition or centroids and there is no efficient method for identifying the initial partitions and the number of clusters.
  - Due to its initial randomness, obtaining the same results on each execution of the algorithm is not guaranteed.
- *k*-means is sensitive to outliers and noise. All objects are forced to belong to one cluster. This would cause the distortion of the recomputed centroid.

1.4 Variants of the *k*-means algorithm

There are some variations of the *k*-means algorithm that solve some of the aforementioned limitations. These are some of them:

- PAM (Kaufman et al., 1990) (partitioning around medoids): This algorithm uses medoids as the cluster prototypes to avoid the effect of outliers. The algorithm is not efficient for large data sets (Han et al., 2001).
- CLARA (Kaufman et al., 1990): Designed to solve the problem of a large data set of PAM.
- ISODATA (iterative self-organizing data analysis technique) (Ball et al., 1965): Employs a technique of merging and splitting clusters, trying to optimize the number of clusters of the result. A cluster is split when its variance is above a pre-specified threshold, and two clusters are merged when its distance is below another pre-specified threshold.
- GKA (genetic-means algorithm) (Krishna et al., 1999): Designed to avoid getting stuck in a local optimum, it can find a global optimum.
- K-modes (Huang, 1998): uses a simple matching coefficient measure to deal with categorical attributes.
- K-prototypes (Huang, 1998): integrates the *k*-means and the *k*-modes algorithms to allow for clustering instances described by mixed attributes.
- X-means (Pelleg et al., 2000): this method automatically finds the number of clusters by using a binary *k*-means, combined with internal validity indices. At each step a *k*-means with $k = 2$ is executed to find a division in two clusters. If the split increases the overall value given by the internal validity indices, the cluster is split and the binary *k*-means continues execution, recursively.
- FW-Kmeans (Feature Weighting *k*-means) (Chan et al., 2004): considers the case with sets of objects that have attributes that are irrelevant. For instance, if the values of an attribute of a set of objects are very different, it can be established that the attribute will not be relevant to form a cluster. So, the most irrelevant attributes of a set (or cluster) have to have also lower weight than the others. In other words, there are attributes that are important to some clusters that are irrelevant to some others. In order to tackle this problem, the following cost function is

proposed: $F(W, Z, \Lambda) = \sum_{l=1}^{k} \sum_{j=1}^{n} \sum_{i=1}^{m} w_{l,j} \lambda_{l,i}^{\beta} d(z_{l,i}, x_{j,i})$, where k is the number of clusters, n is the number of objects, m is the number of attributes, β is an exponent greater than 1, W = $[w_{l,j}]$ is k-by-n integer matrix ($w_{l,j} \in \{0, 1\}$, where 1 indicates that object n belongs to class k), Z contains the cluster centres, Λ = $[\lambda_{l,i}]$ is k-by-m real matrix (the weight of each attribute m for each class k) and $d(z_{l,i}, x_{j,i})$ is the dissimilarity measure between the ith attribute of the centre Z_l and the object X_j. This dissimilarity measure uses the *Euclidean* distance for numerical attributes and the *Hamming* distance for the categorical distance.

- Fuzzy C-means (Song et al., 2007): the conventional clustering approach produces crisp clusters, in which one object can only be assigned to one cluster. However, categorical attributes can often belong to different clusters, because the same word can be applied to different contexts. Moreover, in some real applications, there is often no sharp boundary between clusters. Fuzzy c-means allows assigning a degree of membership to the objects with respect to each of the clusters that are being considered. The fuzzy clustering method partitions the set of objects into k overlapped clusters by considering the following function: $J_m(U, V) = \sum_{c=1}^{K} \sum_{i=1}^{N} U^m(v_c, x_i) d(v_c, x_i)$, where the minimization is performed over all the clusters $v_c \in V$, and $U(v_c, x_i)$ is the membership function for the object x_i belonging to the cluster v_c. To calculate the $d(v_c, x_i)$ the most frequently used approach is the LP norm distance, which is defined as follows (Hathaway et al., 2000): $d(v_c, x_i) = \left(\sum_{j=1}^{S} |x_{i,j} - v_{c,j}|^p \right)^{1/p}$, where p ∈ [1,+∞)and S is the dimensionality of the vectors.

1.5 Some considerations on the main steps of the *k*-means algorithm

The first step of the algorithm, the **initialization**, consists on generating as many clusters as the parameter k, pre-specified by the user. Each cluster is represented by means of a centroid element. The centroid is a representative object that summarizes the values of the members of a given group. So, it is a prototypical object that can be used to know the main characteristics of the objects that belong to the cluster.

The centroid has the same representation format than the rest of objects in the dataset, having the same attributes and taking valid values according to the characteristics of each attribute (i.e. type of values, range, constraints …).

The initialization of the clusters is done by finding k initial centroids, one for each cluster. The determination of appropriate centroids has been studied in the literature (Kaufman et al., 1990; Mirkin, 2005). Three common approaches are the following:

1. A random selection of k objects from the dataset.
2. A guided selection of k objects that are different among them. Some criterion for the selection is required.
3. The user specifies k centroids according to his knowledge of the problem. In this case, the centroids may correspond to one of the objects in the dataset or not.

Once the clustering process starts, at each iteration of the clustering algorithm the objects are placed in different clusters according to their distance (or similarity) to the centroids. The **metrics to measure this**

distance is different depending on the type of values. For numerical values, Euclidean distance is usually applied. For categorical values, the equality/difference of the values is usually considered, as in the Hamming distance. Then, if the value of the object is the same than the value of the centroid (for a given attribute), the distance is 0, otherwise, when they are different, the distance is 1. See some more details about distances in clustering in DAMASK report 3.4.

After generating a partition of the objects in *k* groups, a **new centroid for each cluster** is calculated. As the centroid must represent the "average" value of the attributes, some kind of averaging or aggregation operators are used in this step. For numerical data, the arithmetic average is the most common operator in *k*-means. For categorical attributes, the mode is generally applied.

To include also semantic multi-valued attributes, these three components have been revised and specific methods have been designed. The following section is devoted to the definition and construction of a centroid for multi-valued semantic data.

2 Defining a centroid for semantic multi-valued data

The DAMASK recommender system is based on clustering a set of objects according to their similarity. The similarity is measured taking into account the different types of attributes that describe each object. In the prototype demonstrator that is built in the DAMASK project, the objects are a list of touristic cities that are considered as possible destinations for the users of the recommender system. See DAMASK report 3.2 for more details. So, we will use this case study in this document.

When the attributes take linguistic values with a conceptual interpretation, the previous operators must be changed in order to exploit the semantic component. In this work, we propose different operators that are based on the knowledge represented in ontologies.

In addition, frequently these are multi-valued attributes, which means that a certain object can have more than one value for the attribute. See for example the attribute "Sports" associated to a city (Table 1). The symbol '#' is used as separation mark. In this example, the city of Jerusalem is mainly represented by Basketball and Football, whereas Kunming has much more offer in regards to sports, including Badminton, Tennis or Ice_Hockey.

Table 1. Multi-valued semantic variable

Jerusalem	#Basketball#Football
Kunming	#Basketball#Badminton#Table_Tennis#Tennis#Football#Bowling#Ice_Hockey#Golf#Volleyball
Mexico_City	#Formula_One#Basketball#Football#Ice_Hockey#Golf#Rugby

The centroid of a semantic multi-valued attribute can be represented in three different ways:

1. **Uni-valued**: a single concept is selected to represent all the values of the objects. In this case the most frequent concept on the whole set of terms can be chosen. However, this concept will not represent the rest of values and a lot of information is lost. Another possibility consists on using the taxonomical relations in the ontology to select a concept that is not in the lists of the objects but that is more general and subsumes all or most of them (see the proposal in (Martínez et al., 2012)).

2. **Multi-valued**: a list of concepts is selected according to the lists of each object. The centroid will have the *m* most common concepts among the ones for the attribute in the list of cities of the cluster, where *m* is the mean of the number of concepts for the attribute in the cluster. In this case the new centroid is no different of the other objects. However, we do not have any information about the representativeness of each of the terms that appear in the centroid.

3. **Multi-valued with frequency**: the frequency of appearance of each concept in the cluster's objects is included in the centroid representation; hence, each termin the centroid has a numerical value associated (the frequency) which can be interpreted as the relevance or importance of the concept in the cluster. A minimum threshold to the frequency can be established in order to put a concept in the centroid (as in proposal 2).

2.1 Some approaches to the centroid construction for semantic data

There are two main ways to construct the centroid from in the case of semantic attributes: the one that is based on computing and storing the frequency of appearance of each term or concept, and the one that also use the semantic representation of the values from an ontology.

However, the works dealing with databases rarely consider multi-valued attributes. So, the centroid is a single term that represents best the values of the objects in the cluster. Several methods have been proposed both in the field of Data Mining (or clustering for knowledge discovery) and in the field of Privacy (defining methods to build clusters that are used to mask the data before being released to third parties).

The case of multi-valued semantic attributes is somehow similar to the works dealing with text analysis. Usually documents are summarized using lists of terms. In this framework, there are also some proposals for building lists of representative terms of a set of documents.

Next sections review some recent papers on these two lines.

2.1.1 Frequency-based centroids

Some works consider semantic **attributes in databases** as categorical ones, applying operations based on Boolean (equality/inequality) to compare the terms and on counting the frequency of appearance (i.e. mode). For the case of **uni-valued attributes** we can find several applications using these operators. In (Varde et al., 2006), it is proposed an approach called DesCond to extract a centroid for clusters of scientific input conditions. The centroid is selected from each cluster as a single object (in this case, this refers to all input conditions in a given experiment) such that it is the nearest neighbor to all other objects in the cluster. For this, the centroid is such value in the cluster that the sum of its distances to the rest of values of the cluster is minimal. Because textual attributes are considered as categorical, the distance is defined as 0 if the attribute values are identical and 1 otherwise (Cao et al., 2011; Bai et al., 2011; Domingo-Ferrer et al., 2005; Torra, 2004). In (Torra, 2004; Domingo-Ferrer et al., 2005) authors propose a method for categorical microaggregation of confidential data (i.e., records with values linked to a particular individual) in order to ensure the privacy of individuals before its publication. The microaggregated groups of records are substituted at the end of the algorithm by the centroid of the group. The centroid of textual attributes is selected as the value that most frequently occurs in the group (i.e., mode).

We have found some examples dealing with the case of **multi-valued data** in the field of privacy preservation. In (Erola et al., 2010) authors also use a microaggregation-based masking method to protect query logs, which consist on a list of terms indicated by the user in some search engine to find information in the Web. To group and mask similar queries, it is proposed a clustering algorithm based on finding similarities between queries by exploiting a taxonomy of topics. Then, for each cluster, a centroid consisting of a set of queries replaces all queries in the cluster. Queries in the centroid are selected as those more frequently appearing in the cluster (i.e., mode). In (Greenacre et al., 2010), authors use a similar strategy, classifying documents according to the most frequently appearing words.

The second application domain regards **document analysis**, usually clustering for information retrieval. First we review the case of identifying **a unique value** as centroid representative. In (Bai et al.,

2011), a new method is proposed to find the initial clusters centers for grouping algorithms dealing with categorical data. Authors select the most frequent attribute value (mode) as the cluster representative. In (Cao et al., 2011) it is proposed a dissimilarity measure for clustering categorical objects. Again, the mode is used as the criterion to select cluster representatives. In (Huang et al., 2010) authors proposed a supervised classification algorithm based on labeled training terms and local cluster centers. In order to avoid the interference of mislabeled data, authors select cluster centers so that they reflect the distribution of data (i.e. most frequent labels). In (Ahmed et al., 2005) the authors propose a method capable of dealing with multiple data types when clustering. Each centroid is presented as a vector with mixed types of attributes (numerical and categorical). For the numerical attributes of the centroid, the arithmetic average is user, and for the categorical ones, a frequency-based method. The similarity between objects is computed using a function that calculates each attribute separately. In (Chan et al., 2004) the authors focus on modeling the relevance of each attribute in each cluster. Hence, the authors propose a method to create centroids with weighted attributes and apply low weights to the attributes with low representativeness in the cluster and vice versa. The mode is used as operator for selecting the most appropriate term for each attribute.

Other works consider a list of terms to represent a document, constructing a **multi-valued centroid**. In (Zhang et al., 2010) authors propose to represent the document clusters with a prototype composed by the most frequent terms in a cluster, representing the topic of the grouped documents. In (Song et al., 2007) a method to cluster in a fuzzy manner, making the objects able to belong to more than just one cluster. In order to achieve that, the authors use an Analogue to Language (HAL) model (Lund et al., 1996) as a semantic space model and the fuzzy C-means algorithm (Hathaway et al., 2000). In (Han et al., 2000) a document classification method is introduced. The authors propose a vector of concept frequency for each document, subject to inverse document frequency in order to de-emphasize the concepts with limited discrimination power. To represent a cluster of documents, a concept frequency vector averaging the weights of the various terms present in the documents is created as the cluster's centroid.

2.1.2 Ontology-based centroids

In recent years, some authors started using knowledge sources to assist the construction of centroids. We should distinguish again the case of searching a unique term to represent a set, or a multi-valued list of terms. The works we have found on databases consider only a **uni-valued centroid**, centering the efforts in finding an appropriate term in the ontology to subsume all the ones that appear in the cluster.

The most common approach consists on selecting the Least Common Subsummer (LCS) of the terms, which is the most concrete taxonomical ancestor found in the ontolgy for the terms found in the cluster. For example, in (Abril et al., 2010) authors use the WordNet structured thesaurus (Pedersen et al., 1998) as ontology to assist the classification and masking of confidential textual documents. WordNet models and semantically interlinks more than 100,000 concepts referred by means of English textual labels. Authors exploit WordNet both to assist the classification process, in which relevant words are extracted from text and those are grouped according to the similarity of their meaning, and to select a centroid for each obtained cluster, which is used to mask confidential text. The Wu and Palmer's similarity measure (Wu et al., 1994) is used to estimate the semantic alikeness between words by mapping them to WordNet concepts and computing the number of semantic links separating them. As a result, terms are clusterized according to their semantic similarity. The centroid of the resulting clusters is the LCS. Using this approach, the centroid

represents the semantic content that all the concepts referred in the cluster have in common. Even though term semantics are considered, the use of the LCS as centroid has some drawbacks. First, the presence of outliers (i.e., terms referring to concepts which are semantically far to the major part of the other elements in the cluster) will cause that the LCS becomes a very general concept, for example, in the worst case, the root of the taxonomy. The substitution of cluster terms by such as general concept (e.g., entity, thing, abstraction, etc.) implies a high loss of semantic content. Moreover, the number of term repetitions is not considered during the centroid selection and hence, a scarce term will be considered as important as common ones, biasing results. Those issues imply that the use of the LCS as centroid does not minimize the semantic distance to all elements in the cluster (incoherently to the centroid definition), resulting in a sub-optimal semantic loss.

A more sophisticated approach is proposed in (Guzman-Arenas et al., 2010; Guzmán-Arenas et al., 2011), where the authors introduce the centroid or *consensus* object of a bag of qualitative values. It is commonly assumed that a centroid for a set of qualitative or categorical values is the most popular one, the mode or even the least common ancestor, but the authors try to achieve better results giving a value that minimizes the sum of disagreements for all the objects of a bag (or set) using fuzzy-logic, which is what the article defines as *consensus*. The disagreement when value r is reported instead of the "observer" value s is called the confusion in using r instead of s (Levachkine et al., 2005; Levachkine et al., 2007). The proposal exploits the knowledge modeled in ad-hoc hierarchies that taxonomically link input values to measure the *confusion*. The confusion is computed using the number of descending links in the path from r to s, divided by the height of the hierarchy. However, this method is being affected by the same issue as discussed above; the semantic distance derived from the substitution of a term by its subsumer derives in a noticeable loss of semantic information. Moreover, authors' approach is focused on very simple and overspecified taxonomies that must be constructed ad-hoc for each dataset because they only incorporate the values that appear in the input dataset. Hence, the quality of the results (i.e. the suitability of the selected centroid and the minimization of the semantic distance) closely depends on the homogeneity, completeness and granularity of input values from the taxonomical point of view. In the paper in (Martínez et al., 2012), the authors propose a similar method that can be used in large ontologies. Moreover the frequency of appearance of the terms is combined with the semantic similarity measurement of the centroid candidate terms with respect to the terms that appear in the cluster. This method is explained in the following section.

2.2 A new method for constructing semantic uni-valued centroids

In section 2.1, several approaches to centroid construction for categorical data were discussed. On one hand, centroids computed solely according data distribution, such as the mode, omit the semantics of data and, hence, a crucial dimension of data utility. Moreover, centroids are constrained to values appearing in the input cluster. On the other hand, pure semantic centroids that are based on the ontological concept that subsumes all the values in the cluster, the LCS, are affected by outliers and thus, they commonly suffer from too abstract generalisations. In this section we explain our proposal for building a centroid that takes into account both the frequency of appearance of the values and their semantics.

Let us define a centroid for a set of values as the value (or tuple, in the case of multivariate data) that minimises the distance against all the elements in a set. Formally, given a distance function *d*, the centroid of a set of values $\{v_1, v_2, \ldots, v_n\}$ is defined as:

$$centroid(v_1, v_2, ..., v_n) = arg\ min_c \left\{ \sum_{i=1}^{n} d(c, v_i) \right\} \quad (1)$$

, where *c* is a centroid candidate for the set of arguments.

This definition incorporates the notion of *distance* during the centroid construction. As explained before, semantics should be considered in the distance function *d*, to properly interpret non-numerical concepts so that may preserve, as much as possible, the meaning of original data. For these reason, previously to propose the new centroid method, an appropriate distance measure is required that considers the semantics of concepts during the centroid construction. To interpret data semantics, we will consider semantic similarity measures, which evaluate the taxonomical resemblance of terms according to the knowledge provided by a background ontology. In (Martínez et al., 2012), we take the Wu and Palmer similarity measure and WordNet as the ontology, so that our results can be objectively and unbiasedly compared to related works.

The Wu & Palmer measure evaluates the similarity between two concepts (c_1 and c_2) as the inverse of the number of semantic relationships needed to go from c_1 to c_2 in the background ontology (Eq. 2). This is normalised according to the depth of their Least Common Subsumer (LCS), the most specific ancestor that generalises the two concepts.

$$similarity_{w\&p}(c_1, c_2) \frac{2 \times N_3}{N_1 + N_2 + 2 \times N_3} \quad (2)$$

, where N_1 and N_2 are the number of is-a links (taxonomical relations between specialisations and generalisations from c_1 and c_2, respectively, to their LCS, and N_3 is the number of is-a links from the LCS to the root of the ontology. This ranges from 1 for identical concepts to 0. Hence, this similarity measure is converted into a distance function as follows:

$$dis_{w\&p}(c_1, c_2) = 1 - similarity_{w\&p}(c_1, c_2) \quad (3)$$

The distance measure will be used to assist the centroid considering, the semantics of the data. Moreover, the background knowledge base is exploited not only to assess the semantic distance between terms, but to retrieve centroid candidates.

Semantics should be considered to properly interpret non-numerical concepts so that the centroid may preserve, as much as possible, the meaning of original data. Moreover, data distribution should be taken into account during the centroid selection. To consider the data distribution, we manage the original data set as follows.

Let us take a univariate input cluster with a single categorical attribute *V*. We will represent the information as a tuple of the form: $V = \{\langle v_1, \omega_1\rangle, \ldots, \langle v_p, \omega_p\rangle\}$, where $\langle v_i, w_i\rangle$ define a *value tuple* in which v_i is each distinct term found in the cluster and ω_i is its number of repetitions.

We propose a distance measure that integrates both semantic and distributional aspects to properly evaluate and manage categorical data as follow:

The weighted semantic distance (wsd_O) between two univariate tuples ($\langle v_1, \omega_1\rangle$, $\langle v_2, \omega_2\rangle$), computed from the ontology *O*, is defined as:

$$wsd_O(<v_1,\omega_1>,<v_2,\omega_2>)=\sum_{i=1}^{\omega_1}\sum_{j=1}^{\omega_2} dis_{w\&p}(v_1,v_2)=(\omega_1\cdot\omega_2)\cdot dis_{w\&p}(v_1,v_2) \quad (4)$$

, where the function $dis_{w\&p}$ is the semantic distance expressed in the Eq. 3 (based on Wu & Palmer similarity, Eq. 2) and ω_1 and ω_2 are the number of repetitions of v_1 and v_2 respectively. Note that, to aggregate all individual distance values between elements of value tuples with multiple repetitions (so that data distribution is also considered), their respective appearance frequencies are multiplied.

A second relevant difference of our approach concerns the search space for constructing the centroid. When selecting the centroid according to the frequency of values (e.g. mode), the number of centroid candidates is limited to the set of different values that appear in the cluster. On the contrary, using an ontology such as WordNet, the search space can extended to all concepts modelled in the ontology and hence, the centroid can be constructed from a finer grained set of candidates. The search can be limited to the hierarchical tree to which input values belong, and retrieve some possible centroid candidates as for example sets of taxonomical ancestors. This strategy, combined with the semantic distance defined in Eq. 4, will help to propose more accurate centroids.

First, we formalise our centroid construction method for univariate data.

Let us take $V=\{<v_1,\omega_1>,\ldots,<v_n,\omega_n>\}$ as an input cluster with a single categorical attribute. Let us take an ontology O containing and semantically modelling all v_i in V. The first step of our method consists of mapping the values in V to concepts in O, so that semantically related concepts can be extracted from O following the semantic relationships. We assume that taxonomical subsumers of a term, including itself, are valid representatives of the term. The set of candidates is given in the *minimum subsumer hierarchy*, $H_O(V)$ that goes from the concepts corresponding to the values in V to the Least Common Subsummer of all these values. The *Least Common Subsumer* (LCS) of a set of categorical values V in an ontology O ($LCS_O(V)$) is the deepest taxonomical ancestor that the terms in V have in common for the ontology O. We omitted taxonomical ancestors of the LCS because those will always be more general, that is more semantically distant than the LCS and hence, worse centroid candidates.

The *set of taxonomical subsumers*, $S_{LCSO}(v_i)$ between a certain v_i in V and $LCS_O(V)$ is defined as the set of concepts found in the ontology O that connect via taxonomic relationships v_i and $LCS_O(V)$, including themselves. On ontologies with multiple taxonomical inheritance, several paths can be found between v_i and $LCS_O(V)$; all of them are included in $S_{LCSO}(v_i)$.

The *minimum subsumer hierarchy* ($H_O(V)$) extracted from the ontology O corresponding to all the values in V is defined as the union of all the concepts in $S_{LCS_O}(v_i)$ for all v_i.

$$H_O(V)=\bigcup_{i=1}^{n}\{S_{LCS_O}(v_i)\} \quad (6)$$

, where n is the cardinality of V.

Example 1. As an illustrative example, let us consider a univariate cluster where the attribute refers to the preferred sport: V_1 = {<*boxing*, 1>, <*soccer*, 2>, <*rugby*, 2>, <*contact_sport*, 1>, <*swimming*, 1>,

<*surfing*, 3>}. By mapping these values to concepts found in the background ontology *O* (WordNet) we are able to extract the minimum hierarchy H_{WN}, shown in the Figure 5.

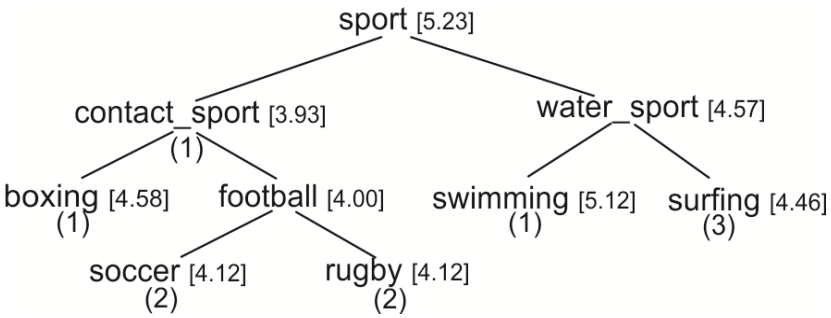


Figure 5: The minimum subsumer hierarchy H_{WN} for the set V_1, extracted from WordNet. Numbers in parenthesis represent the number of repetitions of each value in the cluster. Numbers in brackets represent the accumulated distance of each centroid candidate.

The LCS of the entire set V_1 is $LCS_{WN}(V_1)$=*sport*. For example, the ancestors of *soccer* are S_{LCSWN}(*soccer*)={*soccer*, *football*, *contact_sport*, *sport*}.

All the concepts *c* in H_O are the centroid candidates for *V*. Following Example 1, the centroid candidates of V_1 are those in H_{WN}: {*boxing*, *soccer*, *rugby*, *swimming*, *surfing*, *contact_sport*, *football*, *water_sport*, *sport*}. From the set of centroid candidates, and following the centroid definition (Eq. 1), the term *c* in H_O that minimises the semantic distance to all the v_i in *V* will be selected as the final centroid. In order to consider both semantics and distribution of data, the distance measure presented in Eq. 4 is applied to each centroid candidate.

The centroid of a set of textual values *V* in the ontology *O* is defined as the concept c_j belonging to $H_O(V)$ that minimises the weighted semantic distance wsd_O with respect to all the values of in *V*.

$$centroid_O(V) = \{ argmin(\sum_{i=1}^{n} wsd_O(< c_j, 1 >, < v_i, \omega_i >) \}, \forall c_j \in H_O(V), \forall v_i \ in \ V \qquad (7)$$

If more than one candidate minimises the distance, all of them would be equally representative, and any of them can be selected as the final centroid.

To illustrate the procedure, let us take Example 1. Taking the values in V_1, we obtain the weighted semantic distances for each centroid candidate in H_{WN}. The candidate that minimises the distance against all input values is *contact_sport* with wsd_{WN} (*contact_sport*, V_1) = 3.93. So, this should be taken as the centroid of the set, $centroid_O(V_1)$ = *contact_sport*.

The fact that all the centroid candidates are evaluated to minimise the distance to all values in *V* produces optimal results with respect to the background ontology. It is important to note that, as shown in example, neither the LCS of *V* (*sport*) nor the most frequently appearing value in *V* (*surfing* with 3 appearances) necessarily minimise that distance. In fact, the use of the LCS as centroid for non-uniformly distributed data values, both with respect to their frequency of appearances, but also to their distribution through the hierarchy H_{WN}, typically results in a high semantic distance wsd_{WN}(*sport*, V_1) = 5.23. In this example, the optimal centroid (*contact_sport*) balances the frequency of appearance of the terms and the unbalanced distribution of those terms within the hierarchy, i.e. the cluster has more *contact_sport* branch than *water_sports*.

The method can be generalised for multivariate data considering independent attributes and applying the proposed method individually for each attribute. Note that, in this manner, the centroid construction is optimised at an attribute level, but not at a global level. In this last case, a global centroid selected from the evaluation of all the possible value tuple combinations will be necessary to provide optimal results. However, this will hamper the scalability of our method, requiring the evaluation of an exponentially-large number of value combinations, generated according to the input values and the taxonomical ancestors modelled in the background ontology.

This defintion can be extended to the case of multi-attributes. Then, the centroid of a set of multivariate cluster *MV* in the ontology *O* is defined as:

$$centroid_O(MV) = \{centroid_O(A_1), centroid_O(A_2), ..., centroid_O(A_m)\} \quad (8)$$

, where A_j the set the set of distinct values for the *j*th attribute in *MV* and *m* is the number of attributes in *MV*.

However, for the purpose of the DAMASK project, summarizing all the values into a unique one generates a high loss of information. Our goal is to define a centroid that contains a list of representative terms that summarizes the lists of terms associated to a set of objects. This issue is dealt in the following section.

2.3 The multi-valued semantic-based centroid in DAMASK

The **Multi-valued with frequency** centroid approach is the one selected for the implementation of the recommender system for the DAMASK project, because it permits to have a more complete representation of the clusters. Let us study an example with a cluster with 4 cities. The centroid using the second approach is displayed in Table 2.

Table 2. Cluster example (with the multi-valued centroid at the top)

Centroid	#Football#Basketball#Formula_One#Ice_Hockey#Golf
Jerusalem	#Basketball#Football
Kunming	#Basketball#Badminton#Table_Tennis#Tennis#Football#Bowling#Ice_Hockey#Golf#Volleyball
Madrid	#Formula_One#Basketball#Football#Ballet
Mexico_City	#Formula_One#Basketball#Football#Ice_Hockey#Golf#Rugby

However, considering the frequency of appearance of the terms, we have:

- Football: 4
- Basketball: 4
- Formula_One: 2
- Ice_hockey: 2
- Golf: 2

It can be seen that Football and Basketball are the most frequent concepts in this cluster. Nevertheless, this difference is not represented in the centroid in Table 2. So, the idea is to use this concept count or frequency as a weight for each concept (relevance) in order to improve further calculations, in particular, the

distance between an object and a centroid. Table 3 shows the prototype including the frequency, which is indicated before the concept name.

Table 3. Cluster example (with the multi-valued frequency centroid at the top)

Centroid	#4.Football#4.Basketball#2.Formula_One#2.Ice_Hockey#2.Golf
Jerusalem	#Basketball#Football
Kunming	#Basketball#Badminton#Table_Tennis#Tennis#Football#Bowling#Ice_Hockey#Golf#Volleyball
Madrid	#Formula_One#Basketball#Football#Ballet
Mexico_City	#Formula_One#Basketball#Football#Ice_Hockey#Golf#Rugby

2.3.1 Formalization of the centroid

The centroid of a semantic multi-valued attribute will be represented by a list of tuples of the form a, b . Formally, the centroid c is defined as:

$$c = \quad n_i, t_i \ | n_i > max \ n * \lambda, 1 \quad ,$$

where λ is a threshold to determine the minimum frequency of appearance to be included in the centroid, $n_i$is the number of objects in the cluster that have the term t_i in their description list and n is the overall number of objects in the cluster.

Notice that the purpose of this method is to select only the concepts that appear in a certain percentage of the cities of each cluster. So, $\lambda \in (0, 1]$ that represents the percentage that a concept must appear in the cluster to also appear in the cluster's centroid.

Example. This example illustrates the process for constructing the centroid for a cluster with 4 cities and considering a unique attribute representing the Sport activities in the city, as shown in at the following table:

Table 4. Cities' description in the cluster

Jerusalem	#Basketball#Football
Kunming	#Basketball#Badminton#Table_Tennis#Tennis#Football#Bowling#Ice_Hockey#Golf#Volleyball
Madrid	#Formula_One#Basketball#Football#Ballet
Mexico_City	#Formula_One#Basketball#Football#Ice_Hockey#Golf#Rugby

1. A list with all the concepts appearing in the corresponding attribute in all the cities of the cluster is created:

2. List of terms with its associated frequency of appearance (weight) is created for each attribute of each cluster:

Football	Basketball	Formula_One	Ice_Hockey	Golf	Badminton
4	4	2	2	2	1
Table_Tennis	Tennis	Bowling	Volleyball	Ballet	Rugby
1	1	1	1	1	1

3. A cut over the list of terms in the centroid is done, removing those concepts that are not relevant, so that they are below the value $n * \lambda$, where n is the number of the cities of the cluster and λ is a given threshold.

For example, if we set $= 0.2$, then: $n * \lambda = 4 * 0.2 = 0.8$

Indeed, with this attribute threshold, all the concepts will be accepted for the centroid, even the concepts that only appear once in a cluster. For this reason, we have included in the formulation that the frequency must be always greater than 1. So, after the cut, the centroid is:

Football	Basketball	Formula_One	Ice_Hockey	Golf
4	4	2	2	2

Another example, for a cluster with 14 cities (n = 14) has not this problem of accepting values with frequency equal to one, because $n * \lambda = 14 * 0.2 = 2.8$. So, the centroid in this cluster will have only concepts that appear at least 3 times in its cities.

2.3.2 Normalization of the centroid for clusters comparison

After making the clustering process, we obtain a set of clusters. For each cluster we an construct a semantic multi-valued centroid using the method proposed in the previous sections. However, when the clusters have to be compared with another object, the frequency values included in the centroid are not normalized, giving very different measurement magnitudes for a cluster with 50 objects with regards to another one with 6 objects.

In this section we study how to normalize the weights associated to the terms in the centroid, so that they belong to the interval between 0 and 1.

So, each attribute of each cluster has its correspondent array of concept weights that have to be adjusted

$$W = \ w_1, w_2, \dots, w_n$$

$$W' = f\ W\ = \ w_1', w_2', \dots, w_n' \ ,$$

where W is the array of original weights, w_i is a concept weight (frequency), W' is the array of recomputed weights, w_i' is a recomputed weight and n is the number of concepts that the centroid attribute has.

Three methods are considered for the adjustment of the weights:

- **Common normalization:** this is the most common way to reduce values to a [0, 1] range. It is defined the following way:

$$w_i' = \frac{w_i - \min(W)}{\max W \ - \min(W)}$$

- **Percentage over the sum:** the idea behind this method is to obtain an array of weights that represents the percentage of each concept in relation to the other concepts of the attribute.

  This is the method to obtain the recomputed weights:

$$w_i' = \frac{w_i}{\sum_{j=1}^{n} w_j}$$

- **Percentage over the cluster size:** this method aims to achieve an array of weights that makes each weight a representation of the percentage of appearance of the concept in the cluster.

  This is the formula to obtain the recomputed weights with this method:

$$w_i' = \frac{w_i}{n}$$

These three methods present some different ways to obtain an array of weights that each one is between 0 and 1, but all of them represent different things. For example, let us see its effects on the weights of an example centroid for a cluster with 50 cities:

Football	Basketball	Rugby	Golf	Cricket
35	35	18	15	11

- Common normalization:

Football	Basketball	Rugby	Golf	Cricket
1,00	1,00	0,29	0,17	0,00

- Percentage over the sum:

Football	Basketball	Rugby	Golf	Cricket
0,31	0,31	0,16	0,12	0,10

- Percentage over the cluster size:

Football	Basketball	Rugby	Golf	Cricket
0,70	0,70	0,36	0,30	0,22

It is easy to see that each method returns different results. In the next paragraphs, the characteristics of each method are described.

First, the *common normalization method* gives a result that is not suitable for the purposes of the clustering system, since it is not taking into account the number of no appearances of the concepts in the cluster. For instance, this centroid defines that football appears 35 times in 50 cities. After normalization, its weight value is 1. The same is true for a different cluster centroid that states that football concept appears 50 times in 50 cities. And this is the problem of this method; it is not representing well the weights for the concepts. Another problem is that the concepts that offer the minimum value end up being irrelevant for its 0 weight.

Second, the normalization with *the percentage over the sum* solves the problems of the previous method, and it is interesting because the sum of its values is 1. In fact, this property is not necessary in this case because we will be comparing lists of different lengths, so weights will be used in a different way than the traditional weighted arithmetic operations. Moreover, the value of the weight for one concept depends on the values of the other concepts. Hence, a centroid with high weights for a large number of concepts would result in a recomputed centroid with low weights for its concepts, not representing properly the significance of the term in the cluster.

Third, the *percentage over the cluster size* solves the drawbacks of the two previous methods. Therefore, it has been the selected method to recompute the weights in the clustering system in the DAMASK project. The weights represent the percentage of appearance of the concept in the attribute of the centroid. With this, all the centroids are compared using percentages and not just frequency, solving the aforementioned problems that occur when comparing large clusters with small ones.

2.3.3 Determination of lambda threshold

In the formalization of the centroid section, the lambda threshold was introduced as a value to reduce the number of terms in the cluster centroid. It is worth to remember that this lambda threshold represents the percentage factor that a concept's weight of the centroid must overcome in order to do not be discarded as irrelevant. In this section we study which is the most suitable value for this λ.

The study has been made for the following values of the threshold λ: 0.2, 0.5 and 0.8.

We have made the test with the DAMASK data matrix, which includes 8 semantic attributes. The cities have been grouped in 10 clusters and the centroid for each cluster and attribute has been computed with the method proposed in this document. The number of terms in the centroid is represented in Figure 6.

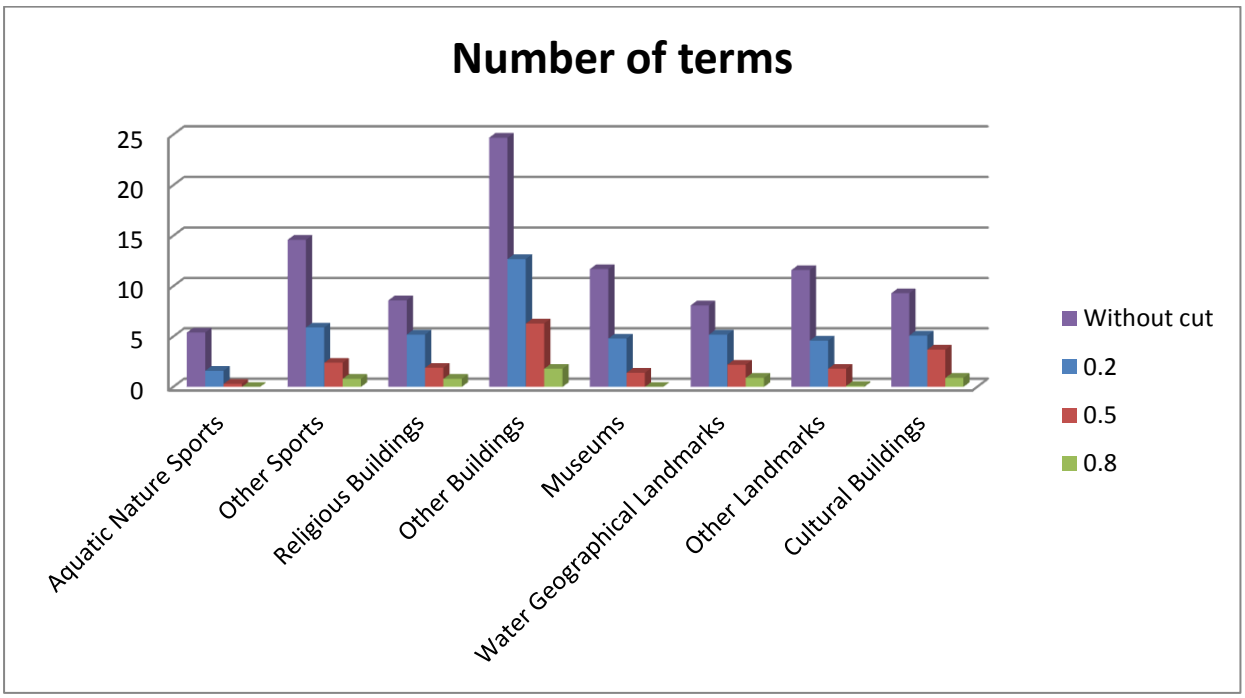


Figure 6: Distribution in the number of terms in the centroid according to different cut thresholds.

For instance, these are the results of the centroid for some attributes:

- *Water geographical landmarks*

-	#12.Beach#31.River#16.Square#19.Hill#4.Terrace#17.Canal#15.Lake#21.Bridge#2.Polder#12.Mountain #3.Stone_Bridge#2.Pedestrian_Bridge#2.Gorge
0.2	#12.Beach#31.River#16.Square#19.Hill#17.Canal#15.Lake#21.Bridge#12.Mountain
0.5	#31.River#19.Hill#21.Bridge
0.8	#31.River

- Museums:

-	#5.Maritime_Museum#22.Art_Gallery#18.Art_Museum#19.Museum#10.Modern_Art_Museum #15.Natural_History_Museum#7.Biographical_Museum#1.Astronomy_Museum #1.Erotic_Museum#4.Railway_Museum#3.Technology_Museum#1.Sex_Museum#1.Woman_Museum #6.Archeology_Museum#2.Music_Museum#2.Toy_Museum#1.Fishing_Museum#3.Industrial_Museum #2.Open_Air_Museum#3.Military_Museum#7.Science_Museum#1.Contemporary_Art_Museum #1.Children_Museum#1.Egyptian_Museum
0.2	#22.Art_Gallery#18.Art_Museum#19.Museum#10.Modern_Art_Museum#15.Natural_History_Museum
0.5	#22.Art_Gallery#19.Museum
0.8	?

- Aquatic Nature Sports

-	#17.Swimming#7.Climbing#19.Cycling#10.Sailing#4.Surfing#5.Skiing#1.Rafting#2.Water_Polo#1.Kayaking#1.Snowboarding#2.Diving#1.Hunting
0.2	#17.Swimming#19.Cycling#10.Sailing
0.5	#19.Cycling
0.8	?

Some interesting results can be seen in these cases. For example, in the *Water geographical landmarks* table, it can be said that a lambda value of 0.8 is too high and results with a representation of the centroid with just one concept. This is absolutely unacceptable for a centroid that originally had 13 concepts and just a few of them are irrelevant at first sight. The results for 0.5 are a bit better; they can be even acceptable since the first discarded concept has a weight of more or less the half of the weight of the most relevant concept. For 0.2, the lowest weight for a concept is 12 which is a good number considering that it is not even a third part of the maximum value of 31. The concepts with irrelevant weights such as 4, 3 or 2 for *terrace*, *stone bridge* or *gorge* between others are discarded.

In Museums the same behavior can be seen, even amplified. Notice that for a λ of 0.8, the centroid results in 0 concepts (marked with the missing symbol ?). For 0.5, just two concepts remain (the original centroid has 24 concepts!), which is unacceptable for the amount of concepts the original centroid has. For 0.2 becomes again the most suitable value for λ for the same reasons as before. With 0.2 the lowest weight is 10, which is more or less the half of the highest, 22. The original centroid has a lot of irrelevant concepts that were cut.

The last example reaffirms what has been seen in the previous examples. For a threshold of 0.8 the result is an empty centroid, 0.5 leaves a centroid unable of represent the cluster, whereas a threshold of 0.2 is gives a more appropriate list of terms.

In conclusion, the values 0.5 or 0.8 remove too many concepts from the centroid. So that, a threshold of $\lambda = 0.2$ seems to be a good value. Consequently this has been the value fixed in the DAMASK system.

Note that changing the λ value would result in notable variations of the clustering result because of the distance algorithm, which is very dependent on concept pairs between the city and the centroid. So, for other applications a similar study should be done in order to find an appropriate threshold for each case.

3 The clustering algorithm, in detail

This section presents the clustering algorithm finally designed and implemented in the DAMASK project. It is an extension of the k-means algorithm that accepts three types of data values: numerical, categorical and semantic. The section is divided into two parts. In the first part, the algorithm is presented. In the second part, each step is explained in detail.

3.1.1 The algorithm

The steps of the *k*-means clustering algorithm presented in section 1.3 have been adapted to deal with objects including numerical, categorical and semantic multi-valued data. The algorithm proceeds as follows:

Determine the number of desired *k* partitions
Start **selecting *k* initial centroids** of differentiated objects
Repeat until there are no changes in the centroids {
    Compute the **distance** of each object to the *k* centroids for numerical, categorical and semantic attributes separately.
    Assign each object to the cluster where its centroid has the lowest distance.
    Compute a new centroid for the computed clusters, for each attribute separately and using a different centroid construction method.
}

The algorithm is the same than the k-means but including different types of operations at some steps of the process. The details about these steps are given in the next section.

3.1.2 Algorithm steps at detail

The three main steps of the clustering algorithm presented are here discussed in more detail.

1. **Select *k* cities as the first centroids:** The *k*-means algorithm has the problem that only finds a local optimum. Because of that, a correct choice for the initial centroids is crucial. The algorithm can also work with random centroids, but for the DAMASK project, a set of 10 initial well differentiated cities has been selected. For this step, the results obtained in a previous work in (Batet et al., 2008) have been used. In that case, a hierarchical clustering method was applied to a smaller set of cities to discover the relations induced by their similarity. Although the set of attributes was slightly different, they also covered numerical, categorical and semantic features not very different from the ones finally used to build the DAMASK data matrix. Therefore, we have considered that the partition obtained in that preliminary work could be used to guide the

selection of the cities. We have taken 10 cities that belong to different clusters of a partition induced by the taxonomical hierarchy obtained in (Batet et al., 2008).

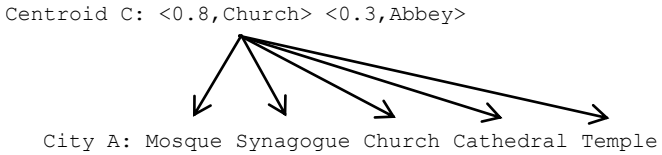
The initial prototypes are then set to: Paris, Barcelona, Krakow, Bangkok, Taipei, Buenos Aires, Havana, Washington D.C., Los Angeles and Abu Dhabi.

Each of these cities has its own values for the attributes considered. Two numerical values for Altitude and Population, two categorical values for the attributes Continent and Climate, and then lists of terms associated to each of the 8 semantic attributes (aquatic nature sports, other sports, religious buildings, cultural buildings, other buildings, museums, water geographical landmarks and other landmarks). For the case of semantic attributes, the centroid must follow the model formalized in section 2.2., as a set of tuples of the form n_i, t_i . Initially the values of n_i are set to 1, so that all tuples are of the form $1, t_i$.

2. The computation of **the distance between a certain city and a certain cluster's centroid** is very similar to the calculation done when comparing two cities (DAMASK report 3.4). For numerical values, the Euclidean distance is used. For categorical data the Hamming distance is applied. For semantic attributes, the measure is based on making an aggregation of partial distance values using the OWA operator. The partial distances are calculated using the Super-Concept based Distance (SCD), which makes an estimation of the distance between two terms based on a ratio of non-common ancestors over the total number of ancestors in a given ontology. The ontology used is again the Tourism ontology, specially designed for this project.

   The difference when comparing an object with a centroid is given by the weight associated to each of the terms in the centroid. This weight is multiplied by the semantic distance calculated with SCD before selecting the minimum value of each pair and applying the OWA operator. By doing this, it is achieved that the most frequent concepts of the cluster are also the more relevant when calculate the distance between a city and the centroid of the cluster.

   For instance, for the attribute "Religious Building" we may have:



   So, the distance between A and C is computed comparing each of the terms in the centroid description with all the terms in the city description and vice versa. Let us take the first comparison, between Church (which appears in 80% of the objects in the cluster) and the values in the city A, obtaining the following array of partial semantic distances: [0.6, 0.6, 0, 0.2, 0.3]. Each of these results is multiplied by the concept weight (for *church*, 0.8): 0.8*[0.6, 0.6, 0, 0.2, 0.3] = [0.48, 0.48, 0, 0.16, 0.24]. The next step consists on selecting the minimum distance to be associated to this pair, in this case is 0, because the city has also a church.

   Let us consider that the city does not have the "church" concept, then, the minimum value would have been 0.16, corresponding to "cathedral". This is coherent with the goal of this

algorithm, because this penalizes the cities that do not have the most relevant concepts of the cluster.

It is easy to see with the previous example, that a city with only a Mosque and a Synagogue will have a large distance to the centroid due to the big weight of a non-similar concept like Church.

The process is then repeated for the Abbey term in comparison with all the concepts in city A, finding a second pair of most similar terms, in this case it would be Abbey and Church.

3. Create new centroids for the computed clusters. This step applies different operators for each type of attribute. For numerical ones, the arithmetic average is used, for categorical, the mode. For semantic multi-valued attributes, the process works just as explained in this document.

4 Implementation

A Java program has been developed to cluster the 150 cities specified in the DAMASK report T3-2. This program follows all the steps presented in section 3.

The program has 2 inputs:

- Excel with the absolute distances (numerical + categorical + semantic) between cities pre-calculated using a little program that follows the specification defined in DAMASK report 3.4.
- An excel file with DAMASK data matrix as defined in DAMASK report 3.2.

The following parameters have been fixed:

- $k = 10$ for the desired number of clusters.
- $\lambda = 0.2$ for the centroid cut process.

The output program is presented to the user with a simple interface presenting the different clusters with their cities:

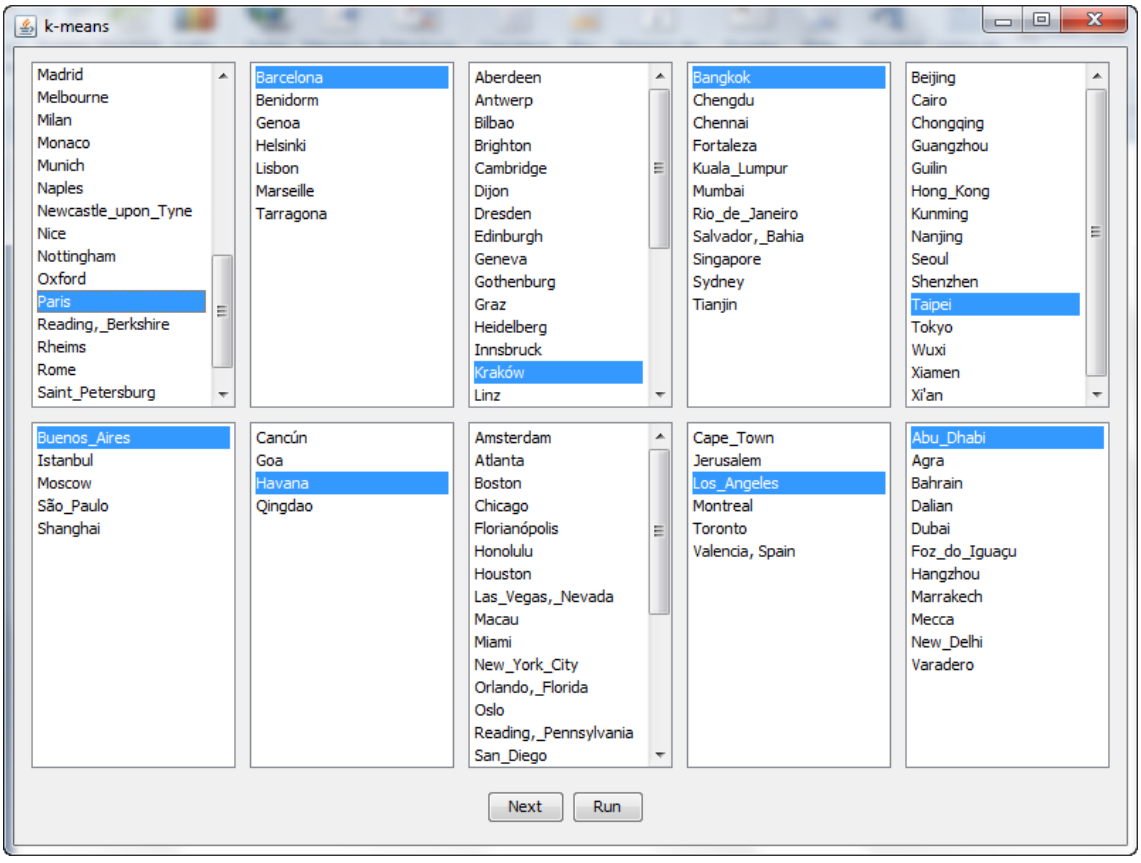


Figure 7: The results window of the program

In Figure 7, we can see the groups of cities obtained. Each group has a city with is highlighted (selected) which indicates the original centroid of the cluster. These will remain selected during all the process to be easy for the user to identify what happens with these cities that are preselected for its dissimilarity all along the process.

The system permits two ways of execution, including two buttons at the bottom of the window. The "Run" button executes the full clustering algorithm. During the execution, the user can see the changes in the clusters in real time. As the process is time consuming due to the assignment of cities, the changes will happen slowly enough for the user to see them on the screen. Despite of that, the button "Next" will allow the user to execute the clustering algorithm step-by-step, what is useful to study the process.

Once the process is finished, a message is shown with the total amount of steps needed to obtain the results.

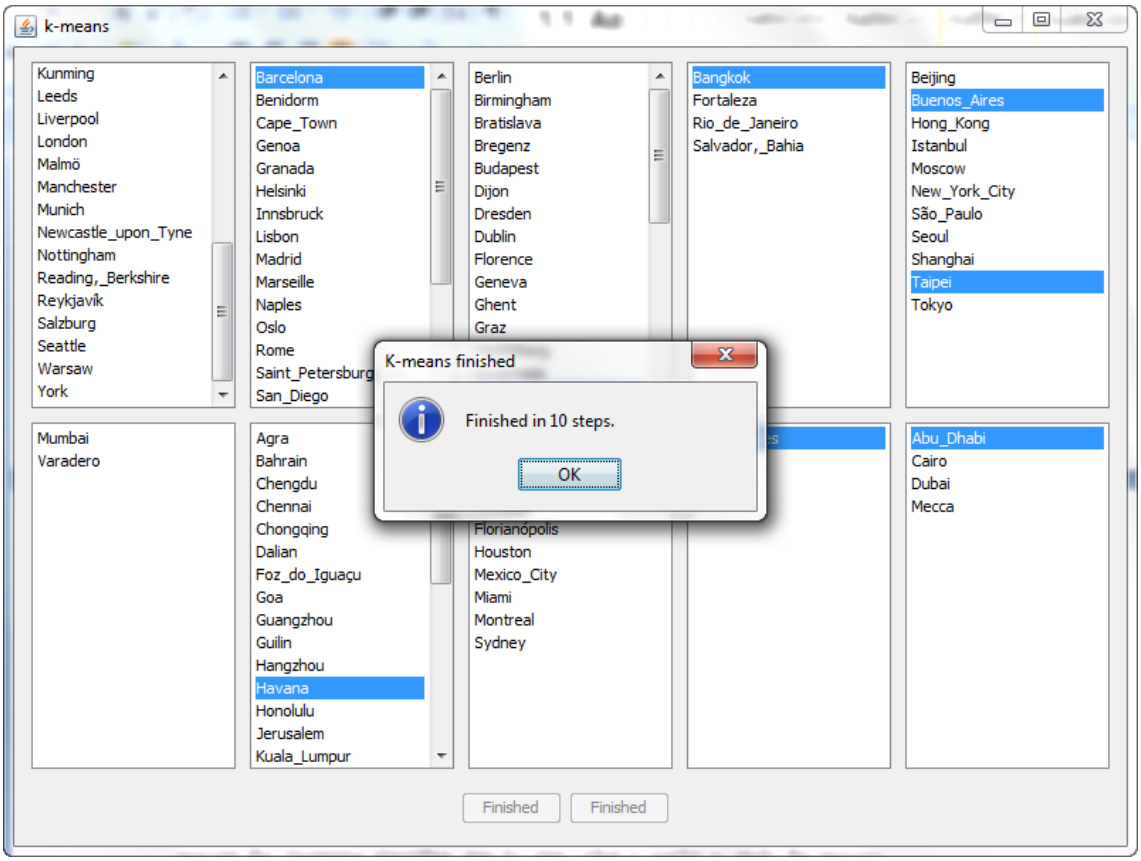
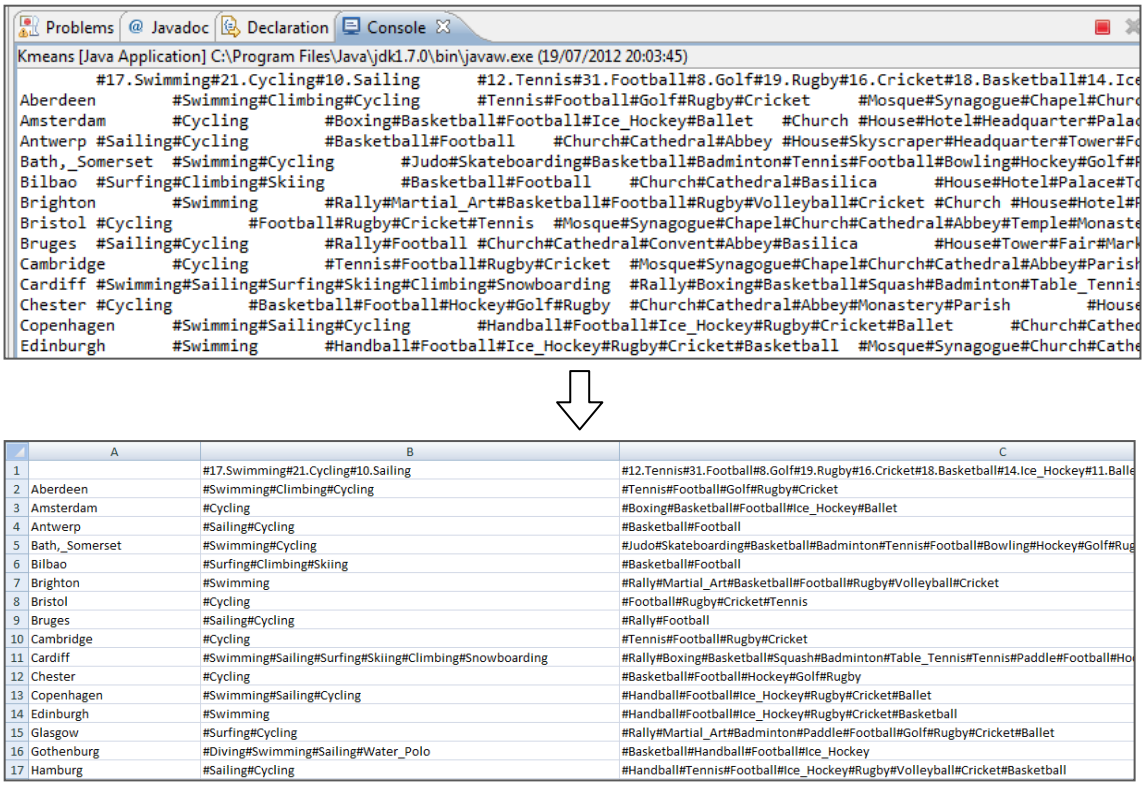


Figure 8: Program announcing the end of the process in X steps.

The results are also printed in the standard output system (the console) with the format required to be copy/pasted to an excel sheet if necessary. This results show a list with all the cities separated in clusters, with all its information as it is in the DAMASK data matrix. At the top of each cluster, it is shown the centroid, along with its concept weights (without normalization). This is an example of the result (just an extract because put here the entire result is impossible for size matters):



Problems | Javadoc | Declaration | Console

Kmeans [Java Application] C:\Program Files\Java\jdk1.7.0\bin\javaw.exe (19/07/2012 20:03:45)

```
        #17.Swimming#21.Cycling#10.Sailing      #12.Tennis#31.Football#8.Golf#19.Rugby#16.Cricket#18.Basketball#14.Ice
Aberdeen        #Swimming#Climbing#Cycling      #Tennis#Football#Golf#Rugby#Cricket     #Mosque#Synagogue#Chapel#Churc
Amsterdam       #Cycling        #Boxing#Basketball#Football#Ice_Hockey#Ballet   #Church #House#Hotel#Headquarter#Palac
Antwerp #Sailing#Cycling        #Basketball#Football    #Church#Cathedral#Abbey #House#Skyscraper#Headquarter#Tower#Fo
Bath,_Somerset  #Swimming#Cycling       #Judo#Skateboarding#Basketball#Badminton#Tennis#Football#Bowling#Hockey#Golf#F
Bilbao  #Surfing#Climbing#Skiing        #Basketball#Football    #Church#Cathedral#Basilica      #House#Hotel#Palace#To
Brighton        #Swimming       #Rally#Martial_Art#Basketball#Football#Rugby#Volleyball#Cricket #Church #House#Hotel#F
Bristol #Cycling        #Football#Rugby#Cricket#Tennis  #Mosque#Synagogue#Chapel#Church#Cathedral#Abbey#Temple#Monaste
Bruges  #Sailing#Cycling        #Rally#Football #Church#Cathedral#Convent#Abbey#Basilica        #House#Tower#Fair#Mark
Cambridge       #Cycling        #Tennis#Football#Rugby#Cricket  #Mosque#Synagogue#Chapel#Church#Cathedral#Abbey#Parish
Cardiff #Swimming#Sailing#Surfing#Skiing#Climbing#Snowboarding  #Rally#Boxing#Basketball#Squash#Badminton#Table_Tennis
Chester #Cycling        #Basketball#Football#Hockey#Golf#Rugby  #Church#Cathedral#Abbey#Monastery#Parish        #House
Copenhagen      #Swimming#Sailing#Cycling       #Handball#Football#Ice_Hockey#Rugby#Cricket#Ballet      #Church#Cathed
Edinburgh       #Swimming       #Handball#Football#Ice_Hockey#Rugby#Cricket#Basketball  #Mosque#Synagogue#Church#Cathe
```

	A	B	C
1		#17.Swimming#21.Cycling#10.Sailing	#12.Tennis#31.Football#8.Golf#19.Rugby#16.Cricket#18.Basketball#14.Ice_Hockey#11.Balle
2	Aberdeen	#Swimming#Climbing#Cycling	#Tennis#Football#Golf#Rugby#Cricket
3	Amsterdam	#Cycling	#Boxing#Basketball#Football#Ice_Hockey#Ballet
4	Antwerp	#Sailing#Cycling	#Basketball#Football
5	Bath,_Somerset	#Swimming#Cycling	#Judo#Skateboarding#Basketball#Badminton#Tennis#Football#Bowling#Hockey#Golf#Rug
6	Bilbao	#Surfing#Climbing#Skiing	#Basketball#Football
7	Brighton	#Swimming	#Rally#Martial_Art#Basketball#Football#Rugby#Volleyball#Cricket
8	Bristol	#Cycling	#Football#Rugby#Cricket#Tennis
9	Bruges	#Sailing#Cycling	#Rally#Football
10	Cambridge	#Cycling	#Tennis#Football#Rugby#Cricket
11	Cardiff	#Swimming#Sailing#Surfing#Skiing#Climbing#Snowboarding	#Rally#Boxing#Basketball#Squash#Badminton#Table_Tennis#Tennis#Paddle#Football#Ho
12	Chester	#Cycling	#Basketball#Football#Hockey#Golf#Rugby
13	Copenhagen	#Swimming#Sailing#Cycling	#Handball#Football#Ice_Hockey#Rugby#Cricket#Ballet
14	Edinburgh	#Swimming	#Handball#Football#Ice_Hockey#Rugby#Cricket#Basketball
15	Glasgow	#Surfing#Cycling	#Rally#Martial_Art#Badminton#Paddle#Football#Golf#Rugby#Cricket#Ballet
16	Gothenburg	#Diving#Swimming#Sailing#Water_Polo	#Basketball#Handball#Football#Ice_Hockey
17	Hamburg	#Sailing#Cycling	#Handball#Tennis#Football#Ice_Hockey#Rugby#Volleyball#Cricket#Basketball

Figure 9: Just with copy/paste, an excel sheet with the results is prepared.

A study of the system along with a survey of the clustering results will be prepared in the DAMASK Deliverable 7.

5 Bibliography

Abril, D. & Navarro-arribas, G. 2010. Towards Semantic Microaggregation of Categorical Data for Confidential Documents. *Proceedings of the 7th international conference on Modeling decisions for artificial intelligence*, 266–276, Springer-Verlag .

Ahmed, R.A., Borah, B., & Bhattacharyya, D.K. 2005. HIMIC : A Hierarchical Mixed Type Data Clustering Algorithm.

Bai, L., Liang, J., & Dang, C. 2011. An initialization method to simultaneously find initial cluster centers and the number of clusters for clustering categorical data. *Knowledge-Based Systems*, 24(6) , 785–795, Elsevier B.V. .

Ball, G.H. & Hall, D.J. 1965. ISODATA, a novel method of data analysis and classification. *Tech Report Stanford University*, Stanford University .

Batet, M., Valls, A., & Gibert, K. 2008. Improving classical clustering with ontologies. *4th World Conference of the International Association for Statistical Computing (IASC)*, (1998) , 137–146.

Cao, F., Liang, J., Li, D., Bai, L., & Dang, C. 2011. A Dissimilarity Measure for the k-Modes Clustering Algorithm. *KNOWLEDGEBASED SYSTEMS*, (July) , Elsevier B.V. .

Chan, E.Y., Ching, W.K., Ng, M.K., & Huang, J.Z. 2004. An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern Recognition*, 37(5) , 943–952.

Domingo-Ferrer, J. & Torra, V. 2005. Ordinal, Continuous and Heterogeneous k-Anonymity Through Microaggregation. *Data Mining and Knowledge Discovery*, 11(2) , 195–212, Kluwer Academic Publishers .

Erola, A., Castella-Roca, J., Navarro-Arribas, G., & Torra, V. 2010. Semantic microaggregation for the anonymization of query logs. *Proceedings of the 2010 international conference on Privacy in statistical databases*, 6344, 127–137, Springer-Verlag .

Forgy, E. 1965. Cluster analysis of multivariate data: Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *Biometrics*, 21, 768–769.

Greenacre, M. & Hastie, T. 2010. Dynamic visualization of statistical learning in the context of high-dimensional textual data. *Web Semantics Science Services and Agents on the World Wide Web*, 8(2-3) , 163–168, Elsevier B.V. .

Gupta, S.K., Rao, K., & Bhatnagar, V. 1999. K-means clustering algorithm for categorical attributes. *DataWarehousing and Knowledge*, 1676/1999, 797.

Guzman-Arenas, A. & Jimenez-Contreras, A. 2010. Obtaining the consensus and inconsistency among a set of assertions on a qualitative attribute. *Expert Systems with Applications*, 37(1) , 158–164.

Guzmán-Arenas, A., Cuevas, A.-D., & Jimenez, A. 2011. The centroid or consensus of a set of objects with qualitative attributes. *Expert Systems with Applications*, 38(5) , 4908–4919, Elsevier Ltd .

Han, E. & Karypis, G. 2000. Centroid-Based Document Classification : Analysis & Experimental Results. *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, 424–431, Springer-Verlag .

Han, J., Kamber, M., & Tung, A. 2001. Spatial Clustering Methods in Data Mining. *Geographic Data Mining and Knowledge Discovery*, Taylor and Francis .

Hansen, P. & Jaumard, B. 1997. Cluster analysis and mathematical programming. *Mathematical Programming*, 79(1-3) , 191–215, Springer .

Hathaway, R.J. & Bezdek, J.C. 2000. Generalized fuzzy c-means clustering strategies using L/sub p/ norm distances. *IEEE Transactions on Fuzzy Systems*, 8(5) , 576–582.

Huang, T., Yu, Y., Guo, G., & Li, K. 2010. A classification algorithm based on local cluster centers with a few labeled training examples. *Knowledge-Based Systems*, 23(6) , 563–571, Elsevier B.V. .

Huang, Z. 1998. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 304, 283–304.

Kaufman, L. & Rousseeuw, P.J. 1990. Finding Groups in Data: An Introduction to Cluster Analysis. *Applied Soft Computing*, 39(1) , 368, John Wiley & Sons .

Krishna, K. & Narasimha Murty, M. 1999. Genetic K-means algorithm. *IEEE transactions on systems man and cybernetics Part B Cybernetics a publication of the IEEE Systems Man and Cybernetics Society*, 29(3) , 433–439, IEEE .

Levachkine, S. & Guzmán-Arenas, A. 2007. Hierarchy as a new data type for qualitative variables. *Expert Systems with Applications*, 32(3) , 899–910.

Levachkine, S., Guzmán-Arenas, A., & Gyves, V.P.D. 2005. The Semantics of Confusion in Hierarchies : Theory and Practice. *Contributions to ICCS 05 13th international conference on cenceptual structures: Common semantics for sharing knowledge*, (Cic) .

Lund, K. & Burgess, C. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2) , 203–208.

MacQueen, J.B. 1967. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1(281-297) , 281–297, University of California Press .

Martínez, S., Valls, A., & Sanchez, D. 2012. Semantically-grounded construction of centroids for datasets with textual attributes. *Knowledge-Based Systems*, 1–26.

Mirkin, B. 2005. Clustering for data mining: a data recovery approach. *Computer Science and Data Analysis Series*, 72(1) , 109, Chapman & Hall/CRC .

Pedersen, T. & Michelizzi, J. 1998. WordNet :: Similarity - Measuring the Relatedness of Concepts. *Architecture*, 21(5) , 38–41, Association for Computational Linguistics .

Pelleg, D. & Moore, A. 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. *Computer*, Seventeent, 727–734, Morgan Kaufmann .

Song, D., Cao, G., Bruza, P., & Lau, R. 2007. Concept Induction via Fuzzy C-means Clustering in a High-dimensional Semantic Space. *Advances in Fuzzy Clustering and its Applications*, John Wiley & Sons, Ltd .

Torra, V. 2004. Microaggregation for Categorical Variables: A Median Based Approach. *Privacy in Statistical Databases*, 3050, 518, Springer Berlin / Heidelberg .

Varde, A.S., Rundensteiner, E.A., Ruiz, C., Brown, D.C., Maniruzzaman, M., & Sisson, R.D. 2006. Designing semantics-preserving cluster representatives for scientific input conditions. *Proceedings of the 15th ACM international conference on Information and knowledge management CIKM 06*, 708–717, ACM Press .

Wu, Z. & Palmer, M. 1994. Verbs semantics and lexical selection. *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* -, 133–138, Association for Computational Linguistics .

Zhang, W., Yoshida, T., Tang, X., & Wang, Q. 2010. Knowledge-Based Systems Text clustering using frequent itemsets. *Knowledge-Based Systems*, 23(5) , 379–388, Elsevier B.V. .