Obtaining general concepts that represent a set of objects using ontologies

Ferran Mata Arcas

Advisor: Aïda Valls Mateu

2012

Master of Science Thesis
Master on Computer Security and Intelligent Systems

Universitat Rovira i Virgili

iTAKA
Intelligent Technologies for Advanced Knowledge Acquisition

Agraïments

Aquest treball ha rebut el suport d'una plaça de Personal d'Administració i Serveis de la Universitat Rovira i Virgili finançada pel *Ministerio de ciència e innovación* per al desenvolupament del projecte DAMASK – Data Mining Algorithms with Semantic Knowledge (TIN2009-11005), junt amb el PlanE (Spanish Economy and Employment Stimulation Plan). El treball s'ha fet dins del grup de recerca ITAKA – Intelligent Technologies for Advanced Knowledge Acquisition.

De la mateixa manera vull agrair a la meva tutora del màster i supervisora d'aquest treball, Aïda Valls, pel seu suport. I als companys del grup de recerca, per l'ambient de treball, l'ajut que era necessari i el bon rotllo.

I finalment a la meva xicota i a la meva família, per la seva paciència i suport en tot moment.

Summary

The amount of information that a user has to deal with nowadays is huge, usually impossible to manage. Because of that that there is a need to reduce the amount of information returned to the user in his searches on the Web. Recommender systems take into account the user's preferences to filter the results and show only a subset of them, the ones relevant for the user. Filtering can be done on the bases of generating a partition of the set of alternatives into clusters. Hence, studying how to generate a general representation of the semantic concepts that represent a certain cluster is needed.

This work is part of the DAMASK (Data Mining Algorithms with Semantic Knowledge) project, and is focused on the last tasks of it. The DAMASK project proposes the use of semantic domain knowledge, represented in the form of ontologies in different tools that are needed to develop Recommender Systems on the Web. The project is conformed in the strategic area of Tourism with the realization of a Web application for the personalised recommendation of touristic destinations. This Master Thesis has been developed within this context. It is based on some previous works done in this research project, mainly focused on the information extraction of relevant data from a domain of structured, semi-structured (e.g. Wikipedia) and unstructured Web resources.

The Master Thesis includes several steps of the DAMASK project. First, the construction of a data matrix that comprehends the cities and their description using a set of heterogeneous attributes. Those attributes have values of multiple types such as semantic, numerical and categorical. This data matrix is built using the data extracted using the tools previously developed in the DAMASK project. Second, there is the need to create and algorithm to compute the similarity between two cities, because one of the main purposes of the DAMASK system is the creation of partitions (clusters) of similar cities, and to do that is necessary a measure of distance. Third, the clustering method chosen to make the partition of cities is the K-means algorithm. At each step of the K-means there is the need to have an average value. In this step, this Master Thesis has developed a new method to generate a multi-valued centroid to represent the semantic attributes of a given cluster. The process of obtaining of general concepts that represent a set of objects is crucial to make these centroids. Using ontologies, we can select the most appropriate and representative concepts for each semantic attribute.

All those techniques are integrated in a Web Recommender System prototype as part of the DAMASK project. The system is designed for non-experienced users with no knowledge on the topic of this work. This recommender system uses the profile of the user that is searching for touristic destinations to show the most appropriate cluster or partition of cities for the user. In other words, the partitions achieved previously are the results that the system can recommend. The user has also the possibility of filter the results if the recommended cluster is too large.

Finally, the results of the clustering system and centroid construction are evaluated separately and in combination with the recommendations that the Web system retrieves to the user. As the reader of this work will see, the results obtained are quite satisfactory.

Index

List of figures

List of tables

1 Introduction

We live in the information society, a world that can make the things easier for the human beings mainly due to the facility of access to information and knowledge. Thanks to the Web 2.0 or the so called Social Web, the amount of information in the Internet has grown exponentially. However, this so large amount of data can also be overwhelming, causing difficulties to manage properly the search of information.

Because of the amount of information, in the last years has emerged the need of having recommenders in the Web. In other words, web applications that autonomously make a selection of a large set of alternatives using the information stored in the user's profile. These tools avoid to the user to have to analyze long lists of alternatives manually in order to search for interesting options that solve his decision problem. For example, this is the case when one is searching for a restaurant, a film or about the next holidays destination.

A recommender system needs to know what the user wants and his interests, his profile. Hence, a user must provide some information about the preferred values for the attributes (i.e. the characteristics) of an object. Once the profile is known, a list or alternatives can be analyzed. We assume that the values of the attributes of these objects can be automatically extracted from the Web using the tools offered by the Semantic Web.

In order to maintain a certain grade of structure on the Web data, the World Wide Web Consortium (W3C) introduced the Semantic Web. According to the W3C, "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries". Tim Berners-Lee, director of the W3C and inventor of the Web defines the Semantic Web as "a Web of data that can be processed directly and indirectly by machines". The Web as we know it is made to be read by humans, and this makes it difficult to be read or interpreted by a machine for tasks such as classifying, searching and acting upon information. Because of that, the purpose of the Semantic Web is to keep a transparent structure to the user to make the information readable by machines.

The Semantic Web involves the use of metadata such as ontologies. The ontologies are formal, explicit specifications of shared conceptualizations. An ontology is a description of the concepts, relationships and even restrictions that can exist for an agent or a community of agents.

To work with large amounts of data, data mining methods are being developed. In particular, clustering techniques such as k-means (Forgy, 1965; MacQueen, 1967) or hierarchical clustering (Calinski et al., 1974) helps us into make groups or subsets of homogeneous data. Clustering techniques permit to represent a large number of objects with a determined, and much smaller, number of clusters that include only similar objects.

In the DAMASK research project (funded by the Spanish government), we assume that, at the top of the set of clusters, a recommender system can be built. The recommendation is done by comparing the user profile with the profile of each cluster (called centroid). In this way, the complexity is reduced because the system does not work with the entire set of objects, but rather on the centroids on a reduced number of clusters.

In this approach, it is of great importance to be able of creating one centroid that represent, as best as possible, a given set of objects, in other words, a cluster. The clustering problem and the generation of centroids have been largely studied for numerical and categorical values. In this master thesis, we will concentrate on another type of values, called semantic attributes, whose values are a list of concepts that can be represented by means of ontologies. Hence, find a way to **obtain general concepts that represent a set of objects using ontologies** is crucial. This problem has been poorly studied up to now, as will be presented in this thesis.

In a further step, we also consider simultaneously different types of attributes to describe the objects. The treatment of heterogeneous data types adds more complexity to the data mining techniques, specially when the majority of these attributes are sets of semantic concepts (multi-valued attributes).

These are some of the goals proposed in the DAMASK research project. The work of this Master Thesis has been devoted in the frame of the DAMASK project, as detailed below.

1.1 The DAMASK project

One of the main limitations of traditional data mining methods is the lack of use of domain knowledge. The DAMASK project proposes the use of semantic domain knowledge, represented in the form of ontologies in different tools that are needed to develop recommender Systems on the Web. More concretely, the project is centred in the application of ontologies to the following aspects:

1. Pre-processing of input data, focusing on their acquisition from freely and massively available resources such as Web resources, their integration and their transformation in a format which may be directly processed.
2. Methods of automatic classification of data, considering any type of heterogeneous information, including numerical, categorical and conceptual data.
3. Methods for interpreting the classes obtained in the previous step.

At the end, a system able to group semantically related Web contents will be developed. As a case of study, the system will be applied to the Tourism domain, concretely to the recommendation of tourism destinations.

1.1.1 Application domain

The methods developed in the DAMASK project will be tested on a recommender system about touristic city destinations.

Information and Communication Technologies (ICTs) applied to fields like Tourism have grown in importance in the last years. In fact, the report about Information Economy of the United Nations Conference on Commerce and Development 2008 (Various, 2008) is centred in the opportunities which ICTs offer in the Tourism field. This shows that it is an important matter not only in the global agenda but also at national and local levels because it stimulates both private and public industries (Villar, 2007).

Different studies indicate that the Web is a great mechanism for tourist destination promotion at all levels (including local one) (Díaz et al., 2000; Díaz et al., 2006). They indicate that consumers all around the world use Internet in order to obtain information about travelling and 54% of them use Internet to initiate the search for travel agencies in the Web. Thus, information about destinations is of crucial importance in decision making in the Tourism field. In consequence, the difficulty of accessing those data may introduce a bias between different destinations. In this sense, one of the most important topics in e-Tourism is to avoid the saturation of common destinations by means of the promotion of new ones, achieving a sustainable tourism (Siorpaes et al., 2006). In order to achieve this goal, it is very important that the user receives all the possible tourist offerings which may be related to his preferences, putting special care on emerging destinations which may remain hidden behind other more typical ones. From the Web point of view, it is fundamental the development of digital content and tools that process that content and provide a direct access to those new destinations.

1.1.2 Goals of the DAMASK project

- O1 Processing and extraction of Web resources based on ontologies.
  - O1.1 Extraction of relevant data from a domain of structured, semi-structured (e.g. Wikipedia) and unstructured Web resources.
  - O1.2 Semantic integration of information in an attribute-value matrix that can be used for further clustering methods.
- O2 Design a clustering method based on ontologies.
  - O2.1 Adaptation of traditional clustering methods to create classifications (trees and partitions) using semantic information.
  - O2.2 Definition of methods to analyse automatically the clusters obtained from O2.1.
- O3 Application and validation of those semantic-based clustering methods (O2.1) on the Tourism domain, particularly to recommend destinations to visit.

1.2 Goals of the Master Thesis

This master thesis starts from what is done in the ITAKA research group on objectives O1 and O2. Specifically, this work explains the implementation of a K-means version using already developed tools previous to O2.1, developing the objectives O2 and O.3.

Figure 1 represents the diagram of tasks of the DAMASK project. This master thesis corresponds to the work done in T2.4, T3.2, T3.4 and T3.5 with the documentation presented as Deliverables D5, D6 and D7.

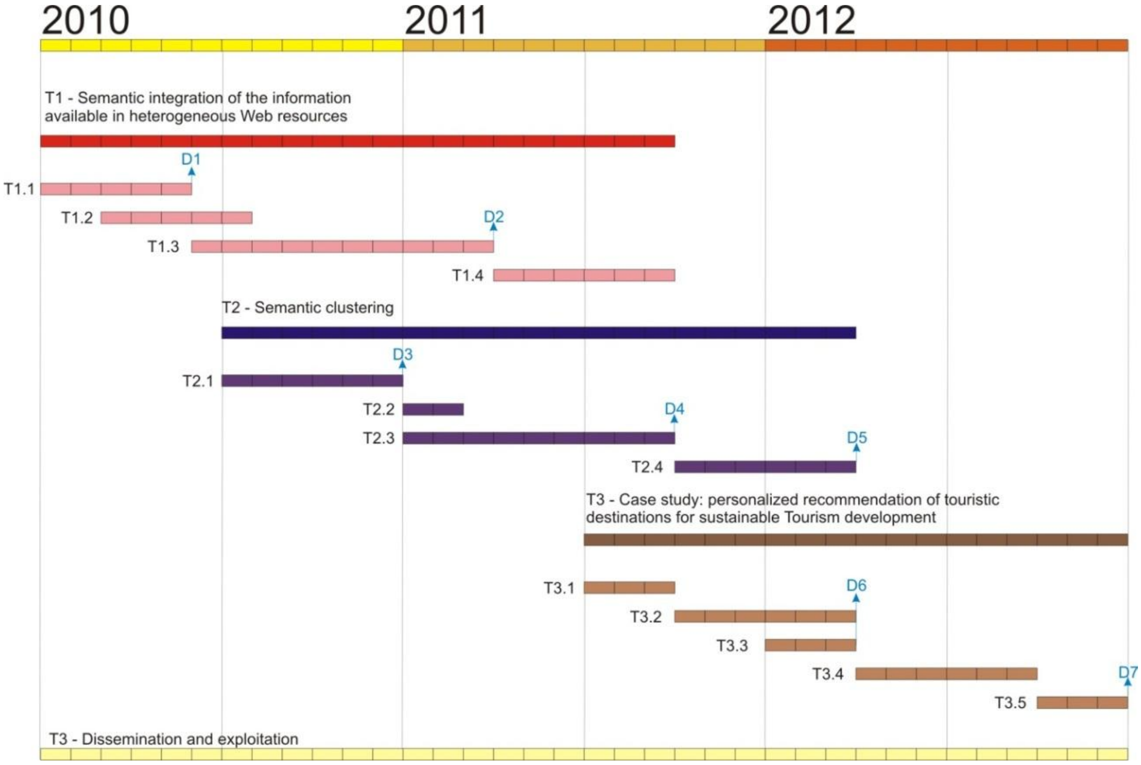


Figure 1: Tasks of DAMASK

The methods developed in tasks T1 (Semantic integration of information from the Web) and T2 (Semantic clustering) are domain-independent. Then, in Task T3, a demonstrator system in the field of Tourism and Leisure is being designed and implemented. In particular, the construction of a Web-based personalized recommender system of touristic destinations for sustainable Tourism development of the most suitable destinations for tourists is studied. Subtask T3-1 was devoted to the creation of a domain ontology specific for this case study. Deliverables D1, D2, D3 and D4 were already done when this master thesis started.

The following are the tasks developed within the context of this master thesis. Subtask T3-2 comprehends the application of the methods of semantic extraction and integration of information (developed in task T1) in order to obtain a data matrix that gathers all the information available in Web resources about some touristic destinations. In T3-4 a method to compute similarity between multi-attribute multi-type objects was developed, in order to be used as the distance values for the K-means implementation in T2-4. The implementation of an adapted K-means algorithm for the multi-attribute multi-type objects (cities) of the DAMASK project was developed in T2-4. After that, a web recommender system was build using the cluster resulting from T2-4. The results of the whole system were analysed and accepted as valid in the last task done within the context of this master thesis. Internal project reports were developed for tasks T3-2 and T3-4, which references the *data matrix construction* and the *distance measures for heterogeneous values* respectively. Deliverables D5, D6 and D7 were presented as the documentation for the *adaptation of the K-means*, the *User-oriented recommender system*, and the *evaluation of the results* respectively.

1.3 Structure of the document

The rest of this document is organized as follows:

- Chapter 2 explains the construction of the data matrix on which the whole system will rely as the base of data. This matrix includes the cities of the system with their attributes and concepts. The tools used to populate this matrix are also explained at detail.
- Chapter 3 presents different ways to compute similarities between cities in a manner to obtain a distance value. This distance value is required for the K-means algorithm.
- Chapter 4 presents the adaptation of the K-means algorithm to work with objects that have several attributes of diverse types. A method to construct centroids that work for each step of the algorithm is also presented.
- Chapter 5 introduces the user-oriented recommender system as a web. It focuses on each step the user has to do to obtain a recommendation and explains each section of the system.
- In Chapter 6, the results of the system are evaluated. It is divided into two main sections, the evaluation of the clustering process explained in chapter 4 and the evaluation of the recommendations given by the web recommender system explained in chapter 5.
- Chapter 7 presents the conclusions of this work among the contributions and the future work.
- Finally, chapter 8 contains the references used in this work.

2 Data Matrix Construction

The goal of the DAMASK (Data Mining Algorithms with Semantic Knowledge) project is the use of semantic domain knowledge, represented in the form of ontologies, to define new methods for extracting and integrating information from heterogeneous Web resources with varying degrees of structure, performing an automatic classification, and making a semantic interpretation of the results.

In previous steps of the DAMASK project, Carlos Vicient (Vicient, 2009; Vicient, 2010) applied methods of semantic extraction and integration of information that were used (and explained in this section) to obtain a data matrix that gathers all the information available in Web resources about some touristic destinations. However, the some treatment of the data obtained has been done to construct the final data matrix that will be used in the recommender system. This chapter explains how the tools developed in the previous tasks of the project have been used and what processes have been done to complete the matrix.

The architecture of the recommender system is given in Figure 2.

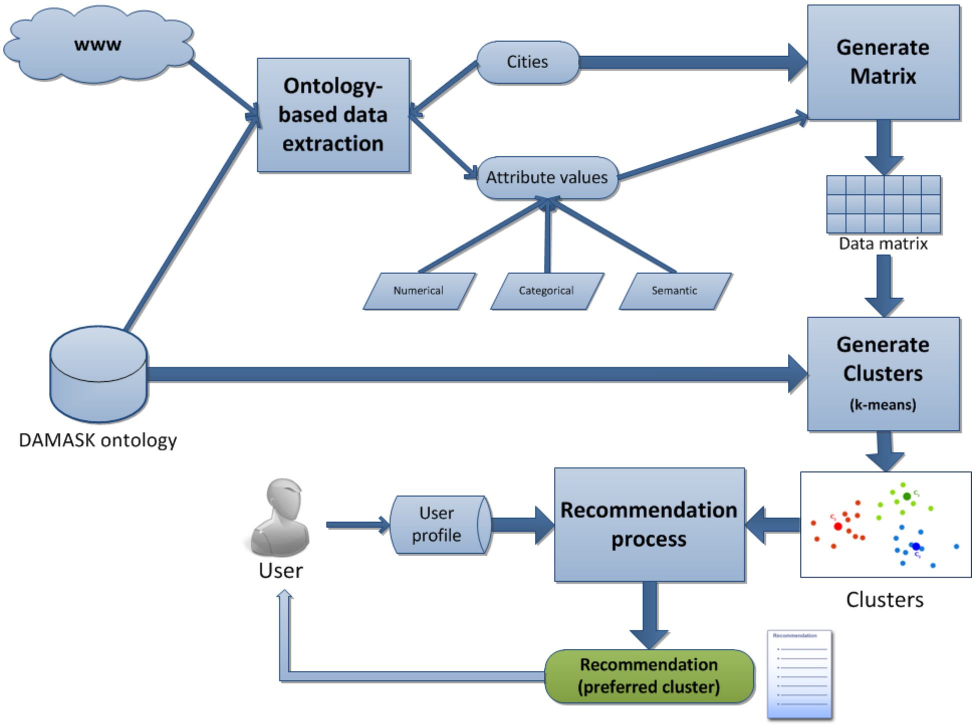


Figure 2: Recommender steps scheme

The process starts with the generation of a data matrix with a diversified sample of cities from all over the world. For each city a set of attributes has been considered. They can be numerical, categorical or semantic. Depending on the nature of the values, different data extraction methods are used to automatically obtain the descriptions of each city from the data available in the Web. In this chapter it is explained how the values of each type of attribute have been obtained.

A set of 150 cities has been selected. These cities are the 150 leading and most dynamic cities in terms of tourist arrivals, according to the ranking made by Euromonitor International in 2006 (Bremner, 2007).

The cities are the following: Aberdeen, Abu Dhabi, Agra, Amsterdam, Antwerp, Atlanta, Bahrain, Bangkok, Barcelona, Bath (Somerset), Beijing, Benidorm, Berlin, Bilbao, Birmingham, Boston, Bratislava, Bregenz, Brighton, Bristol, Bruges, Budapest, Buenos Aires, Cairo, Cambridge, Cancun, Cape Town, Cardiff, Chengdu, Chennai, Chester, Chicago, Chongqing, Copenhagen, Dalian, Dijon, Dresden, Dubai, Dublin, Edinburgh, Florence, Florianopolis, Fortaleza, Foz do Iguaçu, Geneva, Genoa, Ghent, Glasgow, Goa, Gothenburg, Granada, Graz, Guangzhou, Guilin, Hamburg, Hangzhou, Havana, Heidelberg, Helsinki, Hong Kong, Honolulu, Houston, Innsbruck, Inverness, Istanbul, Jerusalem, Krakow, Kuala Lumpur, Kunming, Las Vegas (Nevada), Leeds, Linz, Lisbon, Liverpool, London, Los Angeles, Luxembourg, Lyon, Macau, Madrid, Malmö, Manchester, Marrakech, Marseille, Mecca, Melbourne, Mexico City, Miami, Milan, Monaco, Montreal, Moscow, Mumbai, Munich, Nanjing, Naples, New Delhi, New York City, Newcastle upon Tyne, Nice, Nottingham, Nuremberg, Orlando (Florida), Oslo, Oxford, Paris, Prague, Qingdao, Reading (Berkshire), Reading (Pennsylvania), Reykjavík, Rheims, Rio de Janeiro, Rome, Saint Petersburg, Salvador (Bahia), Salzburg, San Diego, San Francisco, San Jose (California), São Paulo, Seattle, Seoul, Seville, Shanghai, Shenzhen, Singapore, Stockholm, Suzhou, Sydney, Taipei, Tallinn, Tarragona, Tianjin, Tokyo, Toronto, Turku, Valencia, Spain, Varadero, Venice, Vienna, Warsaw, Washington D.C., Wuxi, Xiamen, Xi'an, York, Zaragoza, Zhuhai, Zürich.

2.1 Data matrix by data type

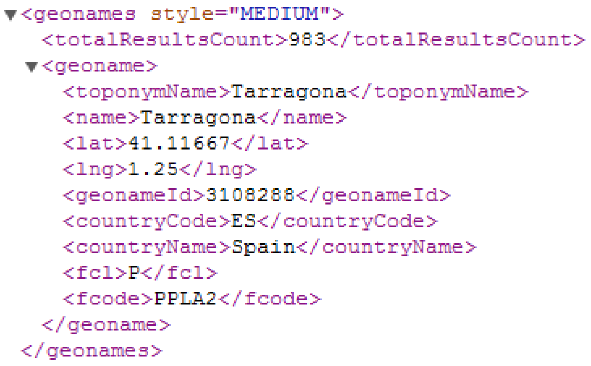
2.1.1 Categorical and numerical data

After the analysis of Web resources made in task T1 of the DAMASK project, the following set of attributes has been selected as significant from a tourist point of view:

- Population – Numerical
- Elevation – Numerical
- Continent code – Categorical
- Climate – Categorical

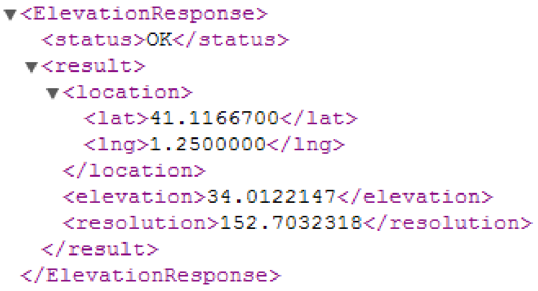
For each city, the *Population* and *Elevation* data were searched by means of the methods developed in task T1, based on using semi-structured resources like Wikipedia in an automatic way (Vicient, 2009). However, for some cities there was information that has been unable to obtain with these techniques (see Deliverable D2 of the DAMASK project for more details).

To complete the data matrix other sources have been considered, such as the use of specific APIs of different Websites. The *population* and *continent code* were extracted from geonames API among many other data that was finally discarded for irrelevant or incomplete. One of the attributes from Geonames that was incomplete is the *elevation* of each city. Then, to get the elevation, the Google Maps Elevation API was used. This API does not return results by city names, but by coordinates instead. These coordinates were extracted from geonames API for each city. So, at the end the *elevation* of the city was found using its coordinates, as shown in Figure 3 and 4.



```
▼<geonames style="MEDIUM">
   <totalResultsCount>983</totalResultsCount>
 ▼<geoname>
     <toponymName>Tarragona</toponymName>
     <name>Tarragona</name>
     <lat>41.11667</lat>
     <lng>1.25</lng>
     <geonameId>3108288</geonameId>
     <countryCode>ES</countryCode>
     <countryName>Spain</countryName>
     <fcl>P</fcl>
     <fcode>PPLA2</fcode>
   </geoname>
 </geonames>
```

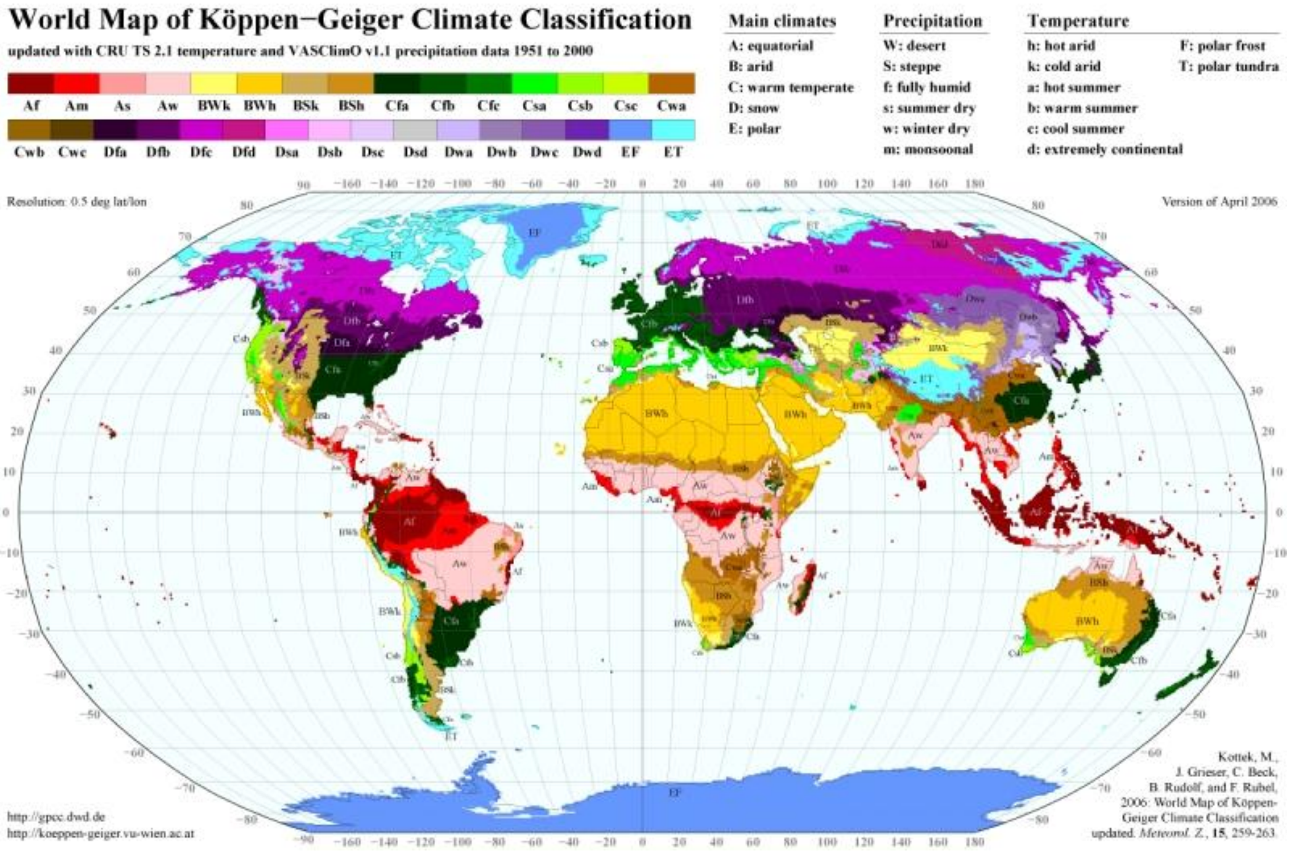
Figure 3: XML result to a Geonames request



```
▼<ElevationResponse>
   <status>OK</status>
 ▼<result>
   ▼<location>
       <lat>41.1166700</lat>
       <lng>1.2500000</lng>
     </location>
     <elevation>34.0122147</elevation>
     <resolution>152.7032318</resolution>
   </result>
 </ElevationResponse>
```

Figure 4: XML result to a G. Maps elevation request

Finally, the *climate* was get from a list available at the Köppen-Geiger website. A file with the correspondences between coordinates in Geonames was used. So, the *climate* for each city was extracted using the Geonames coordinates with the climate information list. The Köppen-Geiger climate classification uses a pattern of characters to indicate the climate of each zone, and those characters indicate the *main climate*, the *precipitation* and the *temperature*. See figure 5 for a global view of the classification.



World Map of Köppen−Geiger Climate Classification
updated with CRU TS 2.1 temperature and VASClimO v1.1 precipitation data 1951 to 2000
Af Am As Aw BWk BWh BSk BSh Cfa Cfb Cfc Csa Csb Csc Cwa
Cwb Cwc Dfa Dfb Dfc Dfd Dsa Dsb Dsc Dsd Dwa Dwb Dwc Dwd EF ET
Main climates
A: equatorial
B: arid
C: warm temperate
D: snow
E: polar
Precipitation
W: desert
S: steppe
f: fully humid
s: summer dry
w: winter dry
m: monsoonal
Temperature
h: hot arid
k: cold arid
a: hot summer
b: warm summer
c: cool summer
d: extremely continental
F: polar frost
T: polar tundra
Resolution: 0.5 deg lat/lon
Version of April 2006
http://gpcc.dwd.de
http://koeppen-geiger.vu-wien.ac.at
Kottek, M., J. Grieser, C. Beck, B. Rudolf, and F. Rubel, 2006: World Map of Köppen-Geiger Climate Classification updated. Meteorol. Z., 15, 259-263.

Figure 5: World map of Köppen-Geiger climate classification

As seen in figure 5, there are 31 possible climates, which is far too much for our sample of 150 cities. So, this classification can be divided into subgroups that are much more adequate for the DAMASK data matrix, and even more semantically coherent. So, the final classification taken is the following:

- Tropical rainforest: Af
- Tropical monsoon: Am
- Tropical savannah: Aw, As
- Desert: BWh, BWk, BWn
- Semi-arid: BSh, BSk
- Humid sub-tropical: Cfa, Cwa
- Oceanic: Cfb, Cwb, Cfc
- Mediterranean: Csa, Csb
- Humid continental: Dfa, Dwa, Dfb, Dwb, Dsa, Dsb
- Subarctic: Dfc, Dwc, Dfd, Dwd, Dsc, Dsd
- Polar: ET, EF

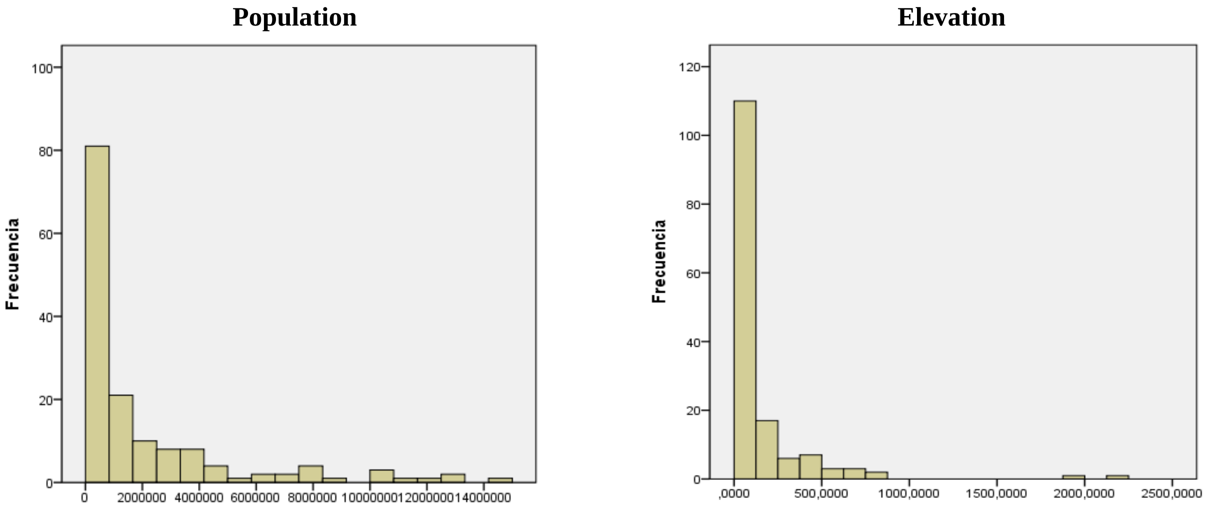
After collecting the information, basic descriptive statistics of the variables have been computed using the SPSS software.

For the numerical values, a few statistics have been calculated and are shown here. These values have been used later in the normalization step performed for calculating distances between cities.

Table 1: Basic descriptive statistics of the numerical variables

Statistics

		Population	Elevation
N		150	150
Avg.		2088459,11	138,915933
Tip. Dev.		3017418,78	283,177017
Min		20000	1,67
Max		14608512	2227,88
Percentiles	25	333414,25	12,605
	50	726002	33,14
	75	2596330,25	156,49


Population

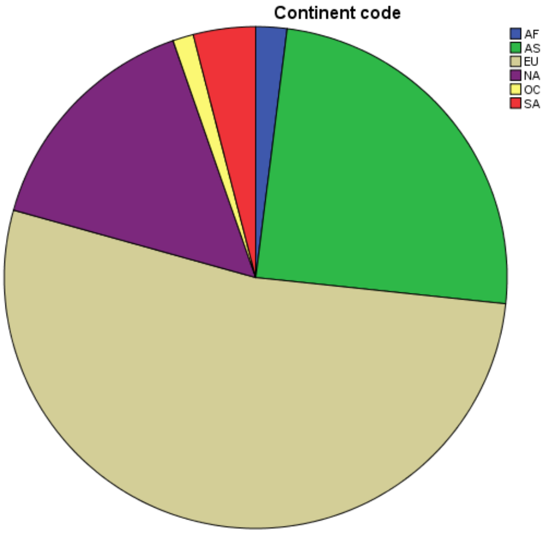
Elevation

Figure 6: Frequency distribution of the numerical variables

This frequency distribution shows a large concentration of values in a very small range of the domain. This indicates that the variables are not very discriminant when used in clustering processes. Small differences must be taken into account, while minimizing the impact of large distances. This consideration has been taken when defining the comparison measure in the clustering process.

Continent code

	Frequency	Percentage	Accumulated
AF	3	2	2
AS	37	24,7	26,7
EU	79	52,7	79,3
NA	23	15,3	94,7
OC	2	1,3	96
SA	6	4	100
Total	150	100	



Climate

	Frequency	Percentage	Accumulated
Desert	6	4	4
Humid continental	15	10	14
Humid sub-tropical	33	22	36
Mediterranean	19	12,7	48,7
Oceanic	60	40	88,7
Semi-arid	2	1,3	90
Subartic	1	0,7	90,7
Tropical monsoon	2	1,3	92
Tropical rainforest	2	1,3	93,3
Tropical savanna	10	6,7	100
Total	150	100	

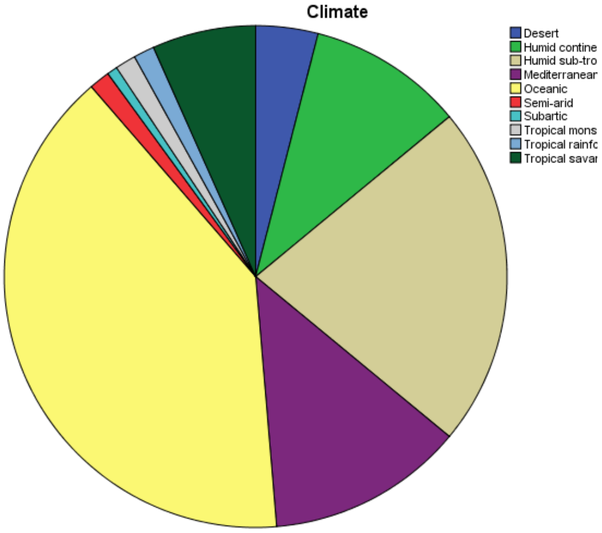


Figure 7: Frequency distribution of the categorical variables

Although we can observe a high concentration of European cities and of Oceanic climate, we consider that the distribution is quite diverse for clustering purposes.

2.1.2 Semantic data

The extraction methods developed in task T1 have also been used to obtain the concepts that correspond to each semantic attribute for a given city. The method is based on the representation of a subset of concepts of the DAMASK ontology that were selected as appropriate to be included as features to describe the touristic destinations. For this purpose, a tailoring of the DAMASK ontology was done, generating a simplified version (see Figure 8). The procedure is explained in (Vicient et al., 2011).

The Damask ontology is the result of merging and combining the following ontologies:

- **TourismOWL.owl:** models touristic points of interest for different kinds of tourist profiles. It was designed in (Vicient, 2009) based on information extracted through Wikipedia articles. It consists of 315 classes and a depth of 5 hierarchical levels. Its main classes represent concepts related with administrative divisions, buildings, festivals, landmarks, museums and sports.

- **Space.owl:** consists of 188 classes and a depth of 6 hierarchical levels. It contains concepts related with three main topics: geographical features, geopolitical entities and places.

- **PCTTO.owl:** It is focused on tourist activities. The ontology represents up to 203 connected concepts in 5 hierarchy levels. It is structured around eight main concepts, which constitute the first level of the hierarchy: “Events”, “Nature”, “Culture”, “Leisure”, “Sports”, “Towns”, “Routes” and “ViewPoints”.

The DAMASK ontology consists of 538 classes connected in 9 hierarchy levels. It is structured around 4 main concepts that constitute the first level of the hierarchy: “geopolitical division”, “activity”, “point of interest” and “geographical feature”. The Damask Ontology is not a pure taxonomy, as it contains multi-inheritance between concepts.

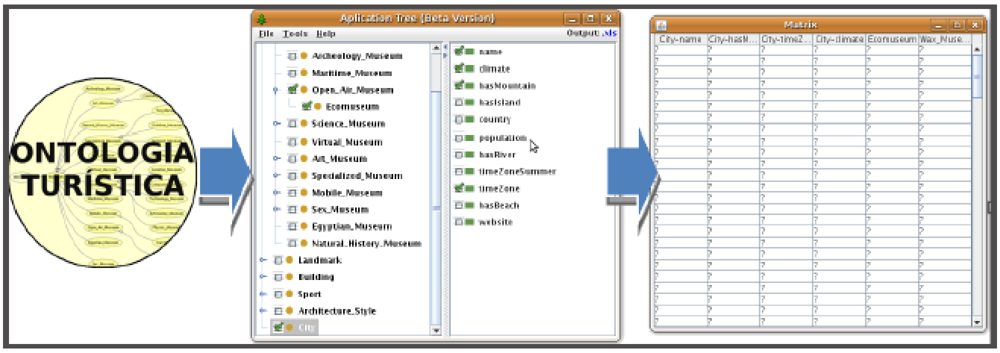


Figure 8: Procedure for the ontology-based feature extraction

A tool for visualizing and manipulating this ontology has been created, named Tree (Figure 9). This tool permits to select a subset of concepts to be used as attributes. If the concept is a leaf on the taxonomy, then the attribute is Binary (a city may have this concept or not). Otherwise the concept generated a "semantic attribute", whose possible values are all the concepts that descend from it.

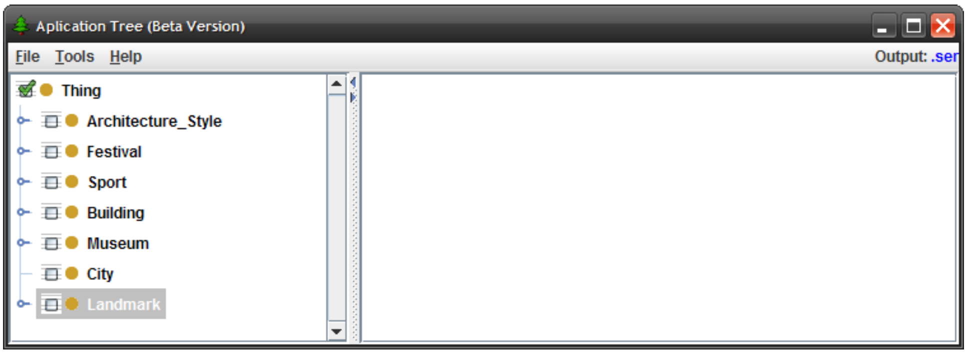


Figure 9: Visualization and manipulation tool for generating the semantic attributes

The main attributes that appear in figure 10 can be unfolded to show subconcepts that can be selected to extract more specific results. The system generates a new attribute for each of the classes (i.e. nodes) selected, and searches if the concepts that are below can be found in the Web pages of each city.

It is crucial to select the most appropriate nodes, at a level that the corresponding attributes that are generated are adequately populated. That means, for instance, that we cannot select categories that are leafs of the taxonomy, because at the lowest level the selected nodes will become a Boolean attribute in the resulting matrix. This type of attribute is not allowed in this system because they provide poor information for making clusters. So, the user must select some of the intermediate nodes of the hierarchy. For example, if *Christian Building* is selected, the resulting attribute will take as values any of its descendants (e.g. chapel, church…), but if the selected node is *Religious Building*, then the resulting attribute can take as values any of its descendants, like chapel, church, mosque, synagogue, Christian building, etc. Notice that the attributes are multi-valued because one city may have more than one Religious Building. The symbol '#' is used to separate the values in each cell of the matrix.

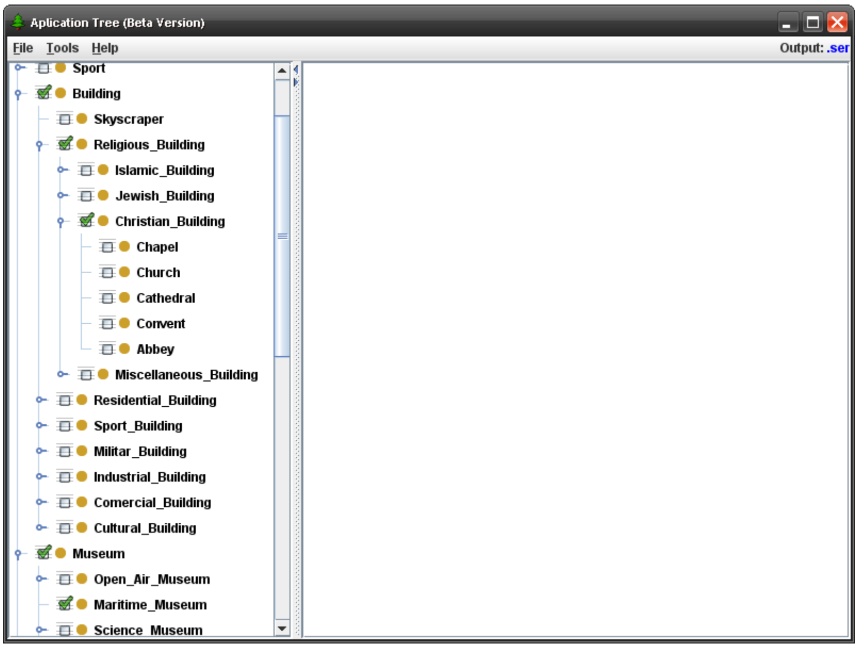
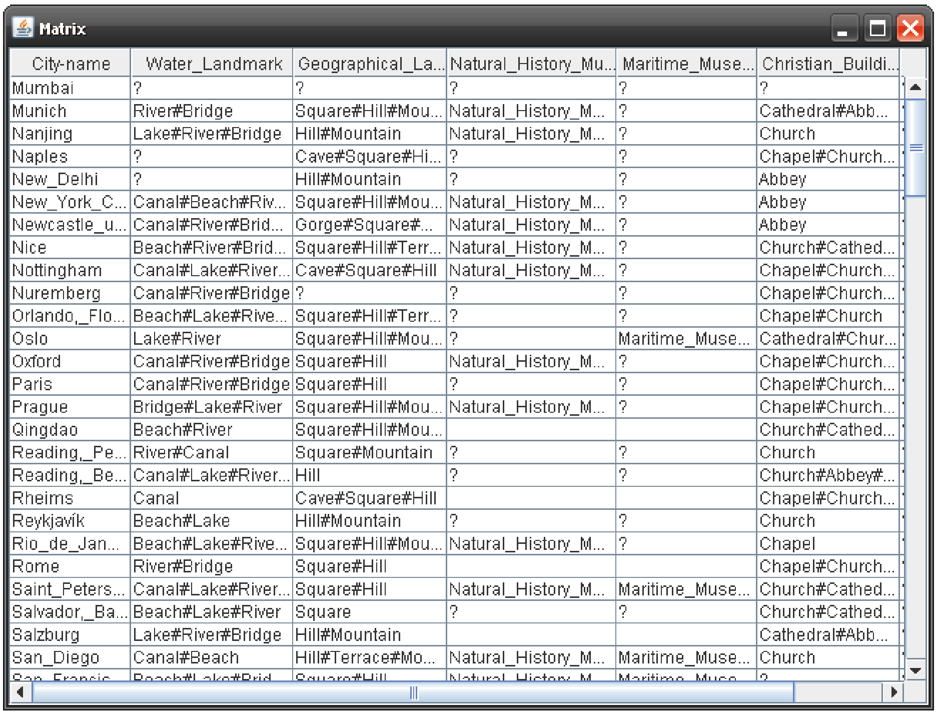


Figure 10: Expanded tree with some unfold categories

In case that the system is not able to find any evidence for a given attribute, the symbol ‘?’ is used. Notice that, in this case, the symbol is not exactly representing a missing value as normally understood, because the lack of information about one attribute is telling us that probably the city does not have any instance of this type. For example, in Figure 11, the automatic extraction system has not found any information about Maritime Museums in Munich, because certainly they do not exist. The data matrix construction by means of extraction processes is slanted by the *precision* and *recall* of each method used during the whole process (i.e., the natural language parser, the named entity detection heuristics, the inaccuracy of Web statistics and the relat-edness measures). The *precision index* measures the number of correct values among all the values obtained. *Recall* is calculated by dividing the number of correct values by the total of values that could have been found. As explained in deliverable D2, for the purpose of the project, high precision is needed, to ensure that the values that we attach to some city are correct. High precision is achieven at a cost of reducing the recall. In this case, the symbol ‘?’ may appear in the data matrix because we have not been able to retrieve the information from the Web page. For example Oslo has a Natural History Museum, but this data has not been found by the system.



City-name	Water_Landmark	Geographical_La...	Natural_History_Mu...	Maritime_Muse...	Christian_Buildi...
Mumbai	?	?	?	?	?
Munich	River#Bridge	Square#Hill#Mou...	Natural_History_M...	?	Cathedral#Abb...
Nanjing	Lake#River#Bridge	Hill#Mountain	Natural_History_M...	?	Church
Naples	?	Cave#Square#Hi...	?	?	Chapel#Church...
New_Delhi	?	Hill#Mountain	?	?	Abbey
New_York_C...	Canal#Beach#Riv...	Square#Hill#Mou...	Natural_History_M...	?	Abbey
Newcastle_u...	Canal#River#Brid...	Gorge#Square#...	Natural_History_M...	?	Abbey
Nice	Beach#River#Brid...	Square#Hill#Terr...	Natural_History_M...	?	Church#Cathed...
Nottingham	Canal#Lake#River...	Cave#Square#Hill	Natural_History_M...	?	Chapel#Church...
Nuremberg	Canal#River#Bridge	?	?	?	Chapel#Church...
Orlando,_Flo...	Beach#Lake#Rive...	Square#Hill#Terr...	?	?	Chapel#Church
Oslo	Lake#River	Square#Hill#Mou...	?	Maritime_Muse...	Cathedral#Chur...
Oxford	Canal#River#Bridge	Square#Hill	Natural_History_M...	?	Chapel#Church...
Paris	Canal#River#Bridge	Square#Hill	?	?	Chapel#Church...
Prague	Bridge#Lake#River	Square#Hill#Mou...	Natural_History_M...	?	Chapel#Church...
Qingdao	Beach#River	Square#Hill#Mou...			Church#Cathed...
Reading,_Pe...	River#Canal	Square#Mountain	?	?	Church
Reading,_Be...	Canal#Lake#River...	Hill	?	?	Church#Abbey#...
Rheims	Canal	Cave#Square#Hill			Chapel#Church...
Reykjavik	Beach#Lake	Hill#Mountain	?	?	Church
Rio_de_Jan...	Beach#Lake#Rive...	Square#Hill#Mou...	Natural_History_M...	?	Chapel
Rome	River#Bridge	Square#Hill			Chapel#Church...
Saint_Peters...	Canal#Lake#River...	Square#Hill	Natural_History_M...	Maritime_Muse...	Church#Cathed...
Salvador,_Ba...	Beach#Lake#River	Square	?	?	Church#Cathed...
Salzburg	Lake#River#Bridge	Hill#Mountain			Cathedral#Abb...
San_Diego	Canal#Beach	Hill#Terrace#Mo...	Natural_History_M...	Maritime_Muse...	Church
San_Francis	Beach#Lake#Brid	Square#Hill	Natural_History_M	Maritime_Muse	?

Figure 11: Resulting matrix from the Tree extracting procedure

2.1.2.1 Selecting the semantic attributes

Since all the concepts in this tailored ontology are candidates to become attributes, we had to select which ones could better represent the city and facilitate the recommendation process for the users. In order to apply the semantic techniques explained in this project, we were not interested in generating Boolean attributes, so the classes at the leaves of the taxonomy where discarded. Then, we selected a subset of the intermediate concepts formed by:

- Aquatic_Sport
- Park
- Nature_Sport
- Martial_Art
- Residential_Building
- Christian_Building
- Water_Landmark
- Militar_Building
- Field_Sport
- Comercial_Building
- Geographical_Landmark
- Sport_Building
- Conmemorate_Landmark
- Cultural_Building
- Memorial_Landmark
- Miscellaneous_Building
- Tomb
- Museum

We studied the number of terms in the lists of each city for the different attributes, as well as, the number of cities with missing information (blanks). Figure 12 shows the average number of terms per attribute (counting only the ones that are not empty). We can see some attributes having a lot of concepts per city and some others practically empty.

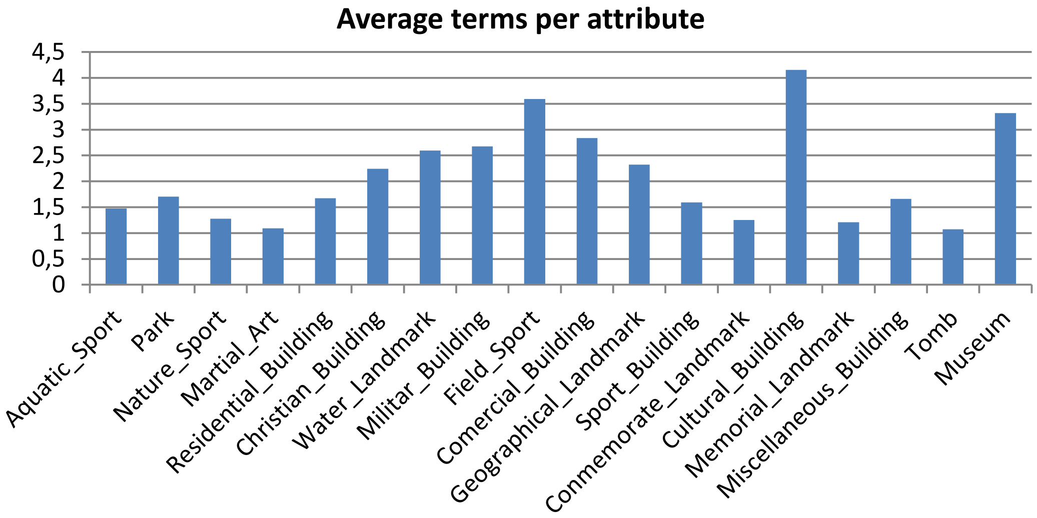


Figure 12: Average number of terms per attribute

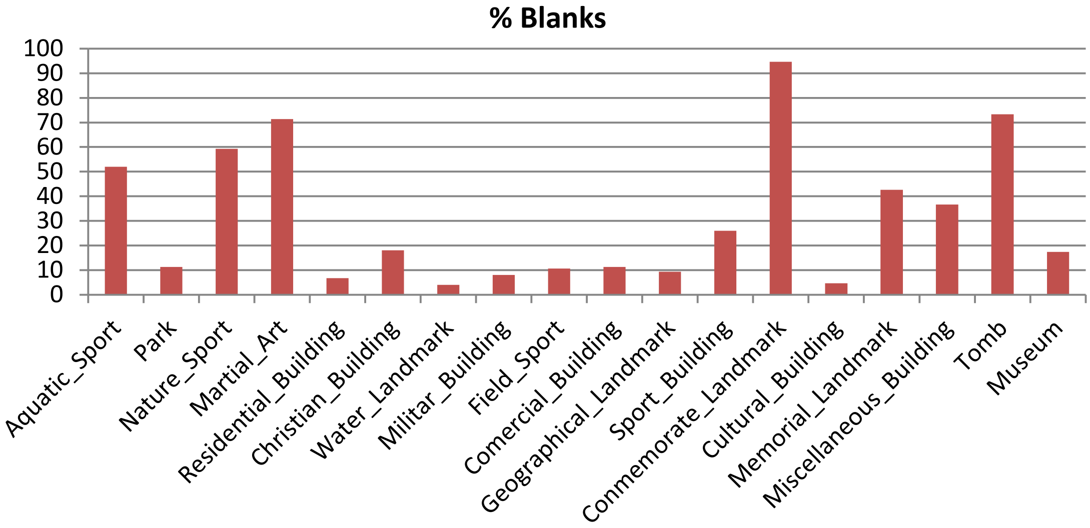
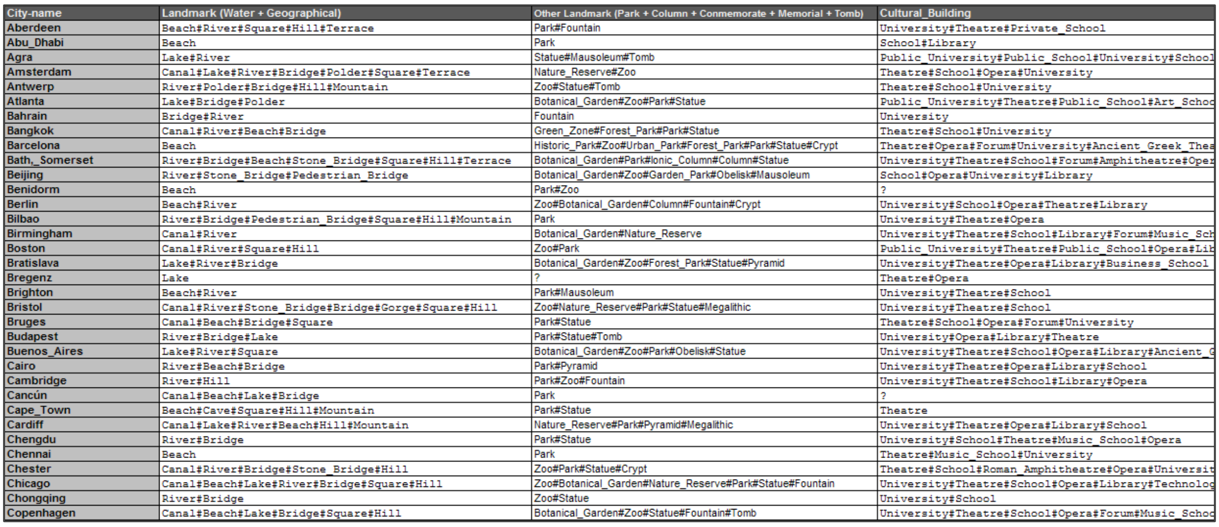


Figure 13: Percentage of blanks (?) per attribute

In Figure 13 we can observe that columns such as *Conmemorate_Landmark* or *Tomb* are mostly empty. This is no desired because these attributes are not able to give any information to distinguish the cities.

After studying this distribution, we decided reduce the number of attributes columns, increasing the terms per attribute and avoiding to have attributes with no information. This is important because it affects the user, who will have to choose a prototype of city considering the meaning of the attributes. Hence, in order to not overwhelm the user, the lower the number of attributes, the better. For example, we constructed a new attributed with Water + Geographical landmarks, and we considered this an important issue for the final data matrix. The final set of attributes is:

- Aquatic + Nature Sports
- Other sports (Motor + Martial + Street + Field + Aerial + Dance + Competition)
- Religious_Building
- Other Buildings (Residential + Skyscraper + Industrial + Militar + Comercial + Sport)
- Museum
- Landmark (Water + Geographical)
- Other Landmark (Park + Column + Conmemorate + Memorial + Tomb)
- Cultural_Building



City-name	Landmark (Water + Geographical)	Other Landmark (Park + Column + Conmemorate + Memorial + Tomb)	Cultural_Building
Aberdeen	Beach#River#Square#Hill#Terrace	Park#Fountain	University#Theatre#Private_School
Abu_Dhabi	Beach	Park	School#Library
Agra	Lake#River	Statue#Mausoleum#Tomb	Public_University#Public_School#University#School
Amsterdam	Canal#Lake#River#Bridge#Polder#Square#Terrace	Nature_Reserve#Zoo	Theatre#School#Opera#University
Antwerp	River#Polder#Bridge#Hill#Mountain	Zoo#Statue#Tomb	Theatre#School#University
Atlanta	Lake#Bridge#Polder	Botanical_Garden#Zoo#Park#Statue	Public_University#Theatre#Public_School#Art_Schoo
Bahrain	Bridge#River	Fountain	University
Bangkok	Canal#River#Beach#Bridge	Green_Zone#Forest_Park#Park#Statue	Theatre#School#University
Barcelona	Beach	Historic_Park#Zoo#Urban_Park#Forest_Park#Park#Statue#Crypt	Theatre#Opera#Forum#University#Ancient_Greek_Thea
Bath,_Somerset	River#Bridge#Beach#Stone_Bridge#Square#Hill#Terrace	Botanical_Garden#Park#Ionic_Column#Column#Statue	University#Theatre#School#Forum#Amphitheatre#Oper
Beijing	River#Stone_Bridge#Pedestrian_Bridge	Botanical_Garden#Zoo#Garden_Park#Obelisk#Mausoleum	School#Opera#University#Library
Benidorm	Beach	Park#Zoo	?
Berlin	Beach#River	Zoo#Botanical_Garden#Column#Fountain#Crypt	University#School#Opera#Theatre#Library
Bilbao	River#Bridge#Pedestrian_Bridge#Square#Hill#Mountain	Park	University#Theatre#Opera
Birmingham	Canal#River	Botanical_Garden#Nature_Reserve	University#Theatre#School#Library#Forum#Music_Sch
Boston	Canal#River#Square#Hill	Zoo#Park	Public_University#Theatre#Public_School#Opera#Lib
Bratislava	Lake#River#Bridge	Botanical_Garden#Zoo#Forest_Park#Statue#Pyramid	University#Theatre#Opera#Library#Business_School
Bregenz	Lake	?	Theatre#Opera
Brighton	Beach#River	Park#Mausoleum	University#Theatre#School
Bristol	Canal#River#Stone_Bridge#Bridge#Gorge#Square#Hill	Zoo#Nature_Reserve#Park#Statue#Megalithic	University#Theatre#School
Bruges	Canal#Beach#Bridge#Square	Park#Statue	Theatre#School#Opera#Forum#University
Budapest	River#Bridge#Lake	Park#Statue#Tomb	University#Opera#Library#Theatre
Buenos_Aires	Lake#River#Square	Botanical_Garden#Zoo#Park#Obelisk#Statue	University#Theatre#School#Opera#Library#Ancient_G
Cairo	River#Beach#Bridge	Park#Pyramid	University#Theatre#Opera#Library#School
Cambridge	River#Hill	Park#Zoo#Fountain	University#Theatre#School#Library#Opera
Cancún	Canal#Beach#Lake#Bridge	Park	?
Cape_Town	Beach#Cave#Square#Hill#Mountain	Park#Statue	Theatre
Cardiff	Canal#Lake#River#Beach#Hill#Mountain	Nature_Reserve#Park#Pyramid#Megalithic	University#Theatre#Opera#Library#School
Chengdu	River#Bridge	Park#Statue	University#School#Theatre#Music_School#Opera
Chennai	Beach	Park	Theatre#Music_School#University
Chester	Canal#River#Bridge#Stone_Bridge#Hill	Zoo#Park#Statue#Crypt	Theatre#School#Roman_Amphitheatre#Opera#Universit
Chicago	Canal#Beach#Lake#River#Bridge#Square#Hill	Zoo#Botanical_Garden#Nature_Reserve#Park#Statue#Fountain	University#Theatre#School#Opera#Library#Technolog
Chongqing	River#Bridge	Zoo#Statue	University#School
Copenhagen	Canal#Beach#Lake#Bridge#Square#Hill	Botanical_Garden#Zoo#Statue#Fountain#Tomb	University#Theatre#School#Opera#Forum#Music_Schoo

Figure 14: Extract of the resulting matrix

Figure 14 shows an extract of three of those attributes. We can see now that the cities have a good number of terms on each of their attributes, having no attributes empty. With this recoding, the number of attributes has been reduced from 18 to 8.

The same analysis was performed in order to prove that now the average of terms per attribute (counting only the ones that are not empty) is acceptable and no column is almost empty. Figures 15 and 16 show the new results:

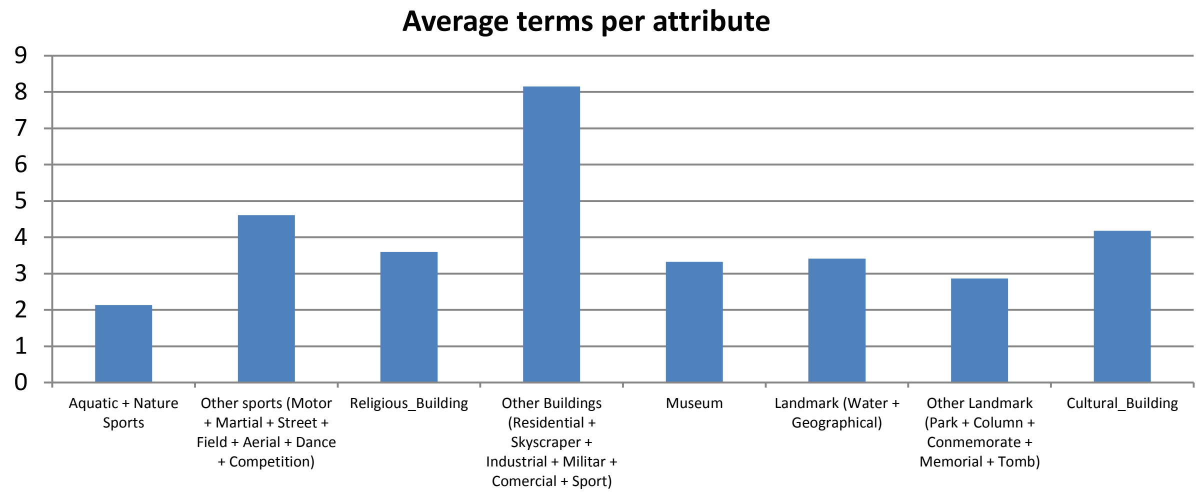


Figure 15: Average number of terms per attribute

We can notice that now almost every attribute has a mean of about 3 or 4 terms per city, which is completely desirable. The counterpart here is that the 'Other Buildings' column is a bit overpopulated. Some possible subdivisions were considered but finally we decided to maintain them as a single group.

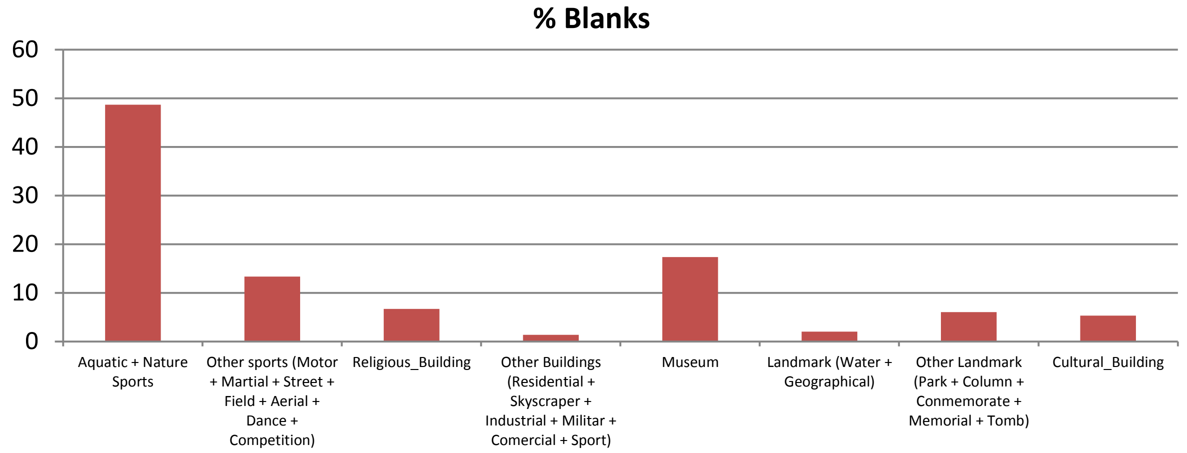


Figure 16: Percentage of blanks (?) per attribute

The percentage of blanks per column shows good results (below 10 – 20 %) in most of the cases. We have an exception in Aquatic + Nature Sports attribute, which has a too large number of missing values, but we decided to maintain it since it has a strong conceptual meaning which is lost if other sports are included in this group.

2.1.3 The DAMASK data matrix

After the study presented in this chapter, the data matrix that will be used in the prototype of the project has the following structure:

1. A set of 150 touristic cities distributed all over the world: Aberdeen, Abu Dhabi, Agra, Amsterdam, Antwerp, Atlanta, Bahrain, Bangkok, Barcelona, Bath (Somerset), Beijing, Benidorm, Berlin, Bilbao, Birmingham, Boston, Bratislava, Bregenz, Brighton, Bristol, Bruges, Budapest, Buenos

Aires, Cairo, Cambridge, Cancun, Cape Town, Cardiff, Chengdu, Chennai, Chester, Chicago, Chongqing, Copenhagen, Dalian, Dijon, Dresden, Dubai, Dublin, Edinburgh, Florence, Florianopolis, Fortaleza, Foz do Iguaçu, Geneva, Genoa, Ghent, Glasgow, Goa, Gothenburg, Granada, Graz, Guangzhou, Guilin, Hamburg, Hangzhou, Havana, Heidelberg, Helsinki, Hong Kong, Honolulu, Houston, Innsbruck, Inverness, Istanbul, Jerusalem, Krakow, Kuala Lumpur, Kunming, Las Vegas (Nevada), Leeds, Linz, Lisbon, Liverpool, London, Los Angeles, Luxembourg, Lyon, Macau, Madrid, Malmö, Manchester, Marrakech, Marseille, Mecca, Melbourne, Mexico City, Miami, Milan, Monaco, Montreal, Moscow, Mumbai, Munich, Nanjing, Naples, New Delhi, New York City, Newcastle upon Tyne, Nice, Nottingham, Nuremberg, Orlando (Florida), Oslo, Oxford, Paris, Prague, Qingdao, Reading (Berkshire), Reading (Pennsylvania), Reykjavík, Rheims, Rio de Janeiro, Rome, Saint Petersburg, Salvador (Bahia), Salzburg, San Diego, San Francisco, San Jose (California), São Paulo, Seattle, Seoul, Seville, Shanghai, Shenzhen, Singapore, Stockholm, Suzhou, Sydney, Taipei, Tallinn, Tarragona, Tianjin, Tokyo, Toronto, Turku, Valencia, Spain, Varadero, Venice, Vienna, Warsaw, Washington D.C., Wuxi, Xiamen, Xi'an, York, Zaragoza, Zhuhai, Zürich.

2. A set of 12 heterogeneous attributes that describe the city with respect to different characteristics that may be useful for the tourist to find the most appropriate destination for his/her holidays. We can divide the attributes in two blocks:
   a) Contextual information:
      - Population (Numerical): to choose small villages or large metropolis
      - Elevation (Numerical): geographical reference of the place (near the sea, in the mountain …)
      - Continent code (Categorical): to restrict the trip to some part of the world
      - Climate (Categorical): general indicator about temperature, humidity

   b) Leisure activities and touristic places:
      - Aquatic and Nature sports
      - Other sports (Motor, martial, street sports, field sports, aerial, dance, competition)
      - Religious buildings that can be visited
      - Cultural buildings
      - Other interesting buildings (Residential, skyscraper, industrial, military, commercial or sport related)
      - Museums
      - Landmarks related to Water and Geographical items
      - Other Landmarks (such as parks, commemorative monuments, memorial monuments, or tombs).

The data matrix is available at http://deim.urv.cat/~itaka/CMS2/index.php

3 Distance Measures for Heterogeneous Values

After deciding the structure of the attributes of the touristic cities, we can study how to compare two cities. A comparison operator is needed to perform any clustering process since the groups of objects are constructed on the basis of the similarity between the objects. A similarity measure for heterogeneous data including multi-valued semantic attributes is proposed in this chapter.

In the previous works of the DAMASK project, Deliverable D3 (Batet et al., 2011), the dissimilarity and distance functions for numerical and categorical data were presented. In Deliverable D4 (Valls et al., 2011) the case of semantic similarity was discussed. Also in D4 a compatibility measure was presented to combine different types of attributes into a single measure. It permits the combination of the contribution of numerical, nominal and semantic features into a global function (Batet, 2010).

In case of not having weights for the different attributes, according to the principles of compatibility measures proposed by Anderberg (Anderberg, 1973), the contribution of a single feature to the final distance can be set up depending on its type and it can be computed per blocks, regarding the types of the considered variables. This expression (Eq. 1) permits to associate a weight to each component, giving different importance to numerical (N), categorical (C) and semantic attributes (S).

$$d(i,i') = \alpha \sum_{k\in N} d_k^N + \frac{\beta}{n_C^2} \sum_{k\in C} d_k^C(i,i') + \frac{\gamma}{n_S^2} \sum_{k\in S} d_k^S(i,i') \qquad (1)$$

In case of knowing the weight that the user wants to give to each type of attribute, the three components in Eq. 1 can be weighted by the user. The set of weights will fulfil that $w_N + w_C + w_S = 1$.

$$d(i,i') = w_N \sum_{k\in N} d_k^N + w_C \sum_{k\in C} d_k^C(i,i') + w_S \sum_{k\in S} d_k^S(i,i') \qquad (2)$$

The quadratic form of the distance $d^2(i,i')$ is required in some clustering methods, such as the Ward criterion that was tested in some previous works (Batet, 2010; Batet et al., 2010). Since in the prototype we are going to apply the k-means algorithm, we do not need a quadratic form, as given in Eq. 2.

In the following sections the details about the distance calculation will be given for each type of attribute.

3.1 Distance calculation

3.1.1 Numerical attributes

In the CITIES data matrix, the numerical attributes are two:

- Population – Numerical
- Elevation – Numerical

The distance will be calculated with the Euclidean distance, as proposed in Deliverable D4. To allow the user to give different overall importance to each feature, we have implemented the weighted Euclidean distance for the attributes in the set *N*. As usual, we consider that $\sum_{k=1}^{|N!} w_k = 1$.

$$d_k^N(i, i') = \sqrt[2]{\sum_{k=1}^{|N|} w_k (x_{ik} - x_{i'k})^2} \tag{3}$$

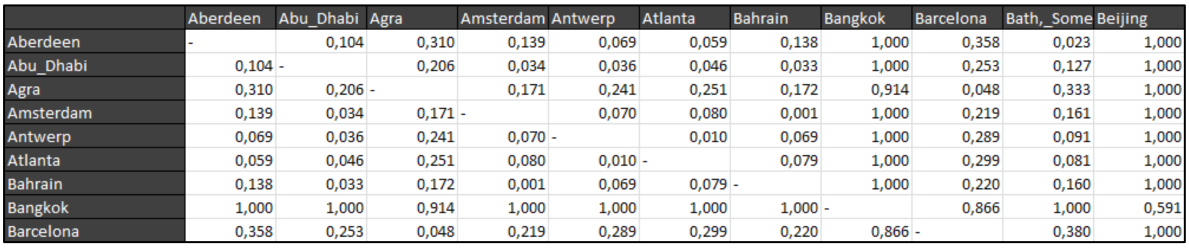
It is worth to note that the values x_{ij} are previously normalized in the range [0, 1] using Eq. 4.

$$x_{norm} = \frac{x - min}{\max - min} \tag{4}$$

Due to the fact that the frequency distribution is not uniform but has a high peak on the low values (see chapter 2), we have taken as maximum value the one at the percentile 85%. Consequently, for the Population attribute, the maximum is fixed at 4,000,000 and cities with highest concentrations of people will receive a normalized value of 1. For the Elevation attribute, the maximum is fixed at 250 meters.

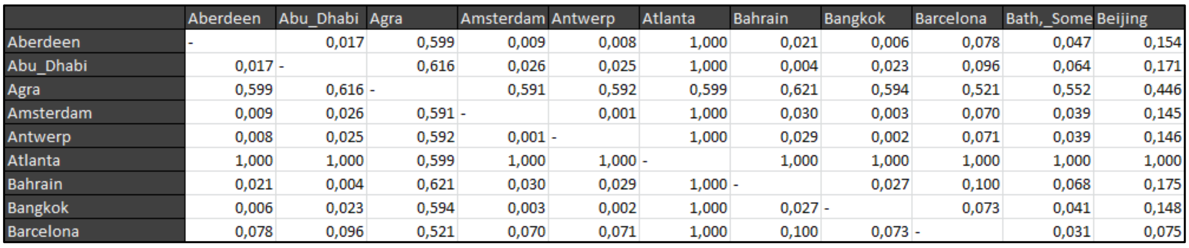
The following tables show some examples of the results obtained.

Table 2: Distances between some cities according to the attribute Population



	Aberdeen	Abu_Dhabi	Agra	Amsterdam	Antwerp	Atlanta	Bahrain	Bangkok	Barcelona	Bath,_Some	Beijing
Aberdeen	-	0,104	0,310	0,139	0,069	0,059	0,138	1,000	0,358	0,023	1,000
Abu_Dhabi	0,104	-	0,206	0,034	0,036	0,046	0,033	1,000	0,253	0,127	1,000
Agra	0,310	0,206	-	0,171	0,241	0,251	0,172	0,914	0,048	0,333	1,000
Amsterdam	0,139	0,034	0,171	-	0,070	0,080	0,001	1,000	0,219	0,161	1,000
Antwerp	0,069	0,036	0,241	0,070	-	0,010	0,069	1,000	0,289	0,091	1,000
Atlanta	0,059	0,046	0,251	0,080	0,010	-	0,079	1,000	0,299	0,081	1,000
Bahrain	0,138	0,033	0,172	0,001	0,069	0,079	-	1,000	0,220	0,160	1,000
Bangkok	1,000	1,000	0,914	1,000	1,000	1,000	1,000	-	0,866	1,000	0,591
Barcelona	0,358	0,253	0,048	0,219	0,289	0,299	0,220	0,866	-	0,380	1,000

Table 3: Distances between some cities according to the attribute Elevation



	Aberdeen	Abu_Dhabi	Agra	Amsterdam	Antwerp	Atlanta	Bahrain	Bangkok	Barcelona	Bath,_Some	Beijing
Aberdeen	-	0,017	0,599	0,009	0,008	1,000	0,021	0,006	0,078	0,047	0,154
Abu_Dhabi	0,017	-	0,616	0,026	0,025	1,000	0,004	0,023	0,096	0,064	0,171
Agra	0,599	0,616	-	0,591	0,592	0,599	0,621	0,594	0,521	0,552	0,446
Amsterdam	0,009	0,026	0,591	-	0,001	1,000	0,030	0,003	0,070	0,039	0,145
Antwerp	0,008	0,025	0,592	0,001	-	1,000	0,029	0,002	0,071	0,039	0,146
Atlanta	1,000	1,000	0,599	1,000	1,000	-	1,000	1,000	1,000	1,000	1,000
Bahrain	0,021	0,004	0,621	0,030	0,029	1,000	-	0,027	0,100	0,068	0,175
Bangkok	0,006	0,023	0,594	0,003	0,002	1,000	0,027	-	0,073	0,041	0,148
Barcelona	0,078	0,096	0,521	0,070	0,071	1,000	0,100	0,073	-	0,031	0,075

3.1.2 Categorical attributes

In the CITIES data matrix, the categorical attributes are two:

- Continent code – Categorical
- Climate – Categorical

In Deliverable D4 the Chi-squared distance is proposed for categorical attributes. This approach considers the frequencies of each category when calculating the distance to another category. Consequently, the underlying distribution of the modalities of the attribute influence the similarity values obtained. In particular, we have used a decomposition of the χ^2 metrics calculation prosed in (Gibert et al., 2003).

$$d_k^2(i,i') = \begin{cases} 0, & if\ x_{ik} = x_{i\prime k} \\ \frac{1}{1_{ki}} + \frac{1}{1_{ki\prime}}, & otherwise \end{cases} \tag{5}$$

Table 4 shows the frequency distribution of the modalities for the two categorical attributes Continent and Climate.

Table 4: Frequency distribution of the Continent and Climate attributes.

Continent	Frequency	%
AF	3	2
AS	37	24,7
EU	79	52,7
NA	23	15,3
OC	2	1,3
SA	6	4

Climate	Frequency	%
Desert	6	4
Humid continental	15	10
Humid sub-tropical	33	22
Mediterranean	19	12,7
Oceanic	60	40
Semi-arid	2	1,3
Subarctic	1	0,7
Tropical monsoon	2	1,3
Tropical rainforest	2	1,3
Tropical savannah	10	6,7

Notice that if we calculate the distance for a city placed in EU and a city placed in SA we obtain:

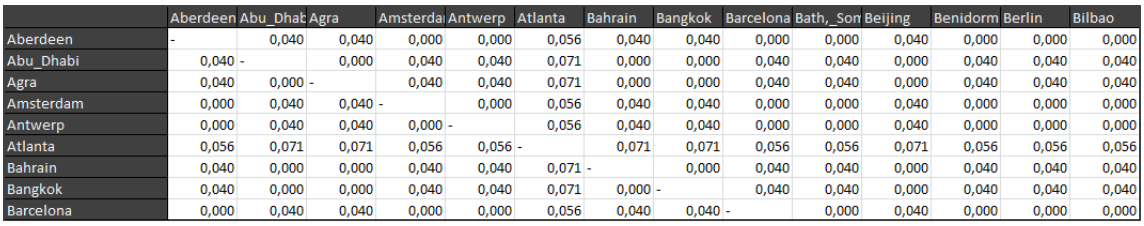
$$d(cA, cB) = \frac{1}{79} + \frac{1}{6} = 0{,}18$$

And a larger distance is obtained when comparing a city placed in AF and another placed in OC:

$$d(cA, cB) = \frac{1}{3} + \frac{1}{2} = 0{,}83$$

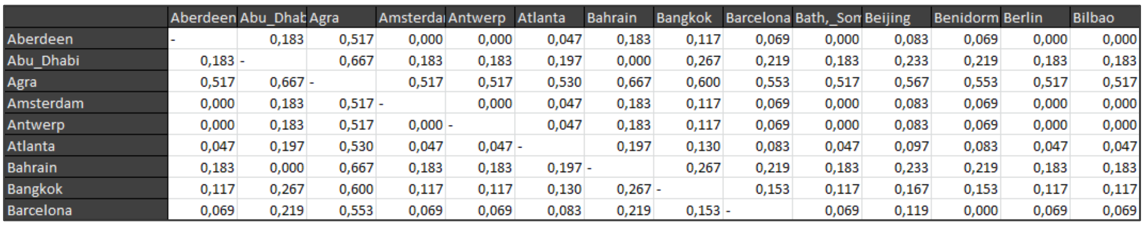
These tables show some examples of the results obtained with the Chi-squared distance:

Table 5: Chi-squared distances between some cities according to the attribute Continent



	Aberdeen	Abu_Dhak	Agra	Amsterdai	Antwerp	Atlanta	Bahrain	Bangkok	Barcelona	Bath,_Son	Beijing	Benidorm	Berlin	Bilbao
Aberdeen	-	0,040	0,040	0,000	0,000	0,056	0,040	0,040	0,000	0,000	0,040	0,000	0,000	0,000
Abu_Dhabi	0,040	-	0,000	0,040	0,040	0,071	0,000	0,000	0,040	0,040	0,000	0,040	0,040	0,040
Agra	0,040	0,000	-	0,040	0,040	0,071	0,000	0,000	0,040	0,040	0,000	0,040	0,040	0,040
Amsterdam	0,000	0,040	0,040	-	0,000	0,056	0,040	0,040	0,000	0,000	0,040	0,000	0,000	0,000
Antwerp	0,000	0,040	0,040	0,000	-	0,056	0,040	0,040	0,000	0,000	0,040	0,000	0,000	0,000
Atlanta	0,056	0,071	0,071	0,056	0,056	-	0,071	0,071	0,056	0,056	0,071	0,056	0,056	0,056
Bahrain	0,040	0,000	0,000	0,040	0,040	0,071	-	0,000	0,040	0,040	0,000	0,040	0,040	0,040
Bangkok	0,040	0,000	0,000	0,040	0,040	0,071	0,000	-	0,040	0,040	0,000	0,040	0,040	0,040
Barcelona	0,000	0,040	0,040	0,000	0,000	0,056	0,040	0,040	-	0,000	0,040	0,000	0,000	0,000

Table 6: Chi-squared distances between some cities according to the attribute



	Aberdeen	Abu_Dhak	Agra	Amsterdai	Antwerp	Atlanta	Bahrain	Bangkok	Barcelona	Bath,_Son	Beijing	Benidorm	Berlin	Bilbao
Aberdeen	-	0,183	0,517	0,000	0,000	0,047	0,183	0,117	0,069	0,000	0,083	0,069	0,000	0,000
Abu_Dhabi	0,183	-	0,667	0,183	0,183	0,197	0,000	0,267	0,219	0,183	0,233	0,219	0,183	0,183
Agra	0,517	0,667	-	0,517	0,517	0,530	0,667	0,600	0,553	0,517	0,567	0,553	0,517	0,517
Amsterdam	0,000	0,183	0,517	-	0,000	0,047	0,183	0,117	0,069	0,000	0,083	0,069	0,000	0,000
Antwerp	0,000	0,183	0,517	0,000	-	0,047	0,183	0,117	0,069	0,000	0,083	0,069	0,000	0,000
Atlanta	0,047	0,197	0,530	0,047	0,047	-	0,197	0,130	0,083	0,047	0,097	0,083	0,047	0,047
Bahrain	0,183	0,000	0,667	0,183	0,183	0,197	-	0,267	0,219	0,183	0,233	0,219	0,183	0,183
Bangkok	0,117	0,267	0,600	0,117	0,117	0,130	0,267	-	0,153	0,117	0,167	0,153	0,117	0,117
Barcelona	0,069	0,219	0,553	0,069	0,069	0,083	0,219	0,153	-	0,069	0,119	0,000	0,069	0,069

We can observe some "attraction" behaviour of the continents with high frequency. The goal of DAMASK recommender system is to help to diversify the destinations that are proposed to a tourist, so it seems not adequate that the cities that are in modalities with high concentration of options increase the similarity among them. For this reason, we finally have taken the Hamming distance based on the equality/inequality of the modalities. This distance takes into account the number of between the differences in the values of the categorical attributes, giving one of the following values when comparing the two modalities of the objects *i* and *i'* for the *k*-th attribute:

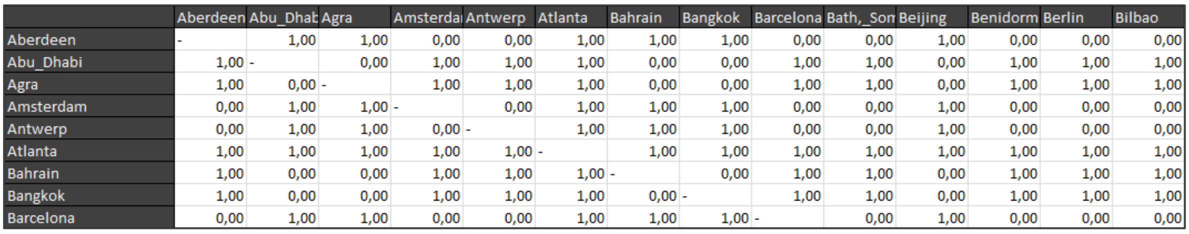
$$d'_k\,(i,i') = \begin{cases} 0 & if\ x_{ik} = x_{i'k} \\ 1 & if\ x_{ik} \neq x_{i'k} \end{cases} \tag{6}$$

Therefore, to calculate the overall distance for the set of categorical variables C, we make a weighted average of the partial distances given by Eq. 6 for each individual attribute, with $\sum_{k=1}^{|C!} w_k = 1$, as defined in Eq. 7.

$$d_k^C(i,i') = \sum_{k=1}^{|C|} w_k d'_k \tag{7}$$

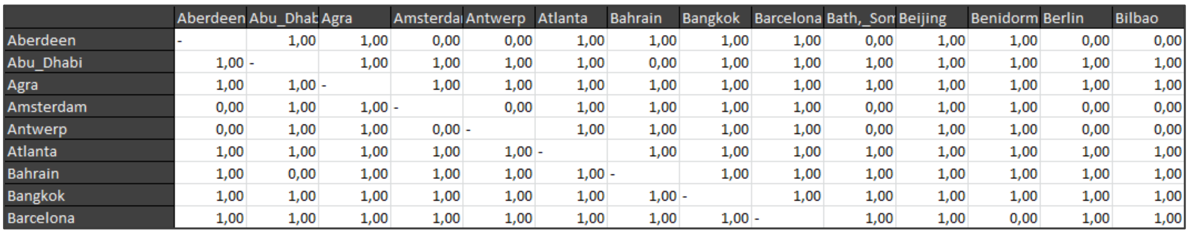
The following tables show some examples of the results obtained with the Hamming distance.

Table 7: Hamming distance between some cities according to the attribute Continent



	Aberdeen	Abu_Dhab	Agra	Amsterda	Antwerp	Atlanta	Bahrain	Bangkok	Barcelona	Bath,_Som	Beijing	Benidorm	Berlin	Bilbao
Aberdeen	-	1,00	1,00	0,00	0,00	1,00	1,00	1,00	0,00	0,00	1,00	0,00	0,00	0,00
Abu_Dhabi	1,00	-	0,00	1,00	1,00	1,00	0,00	0,00	1,00	1,00	0,00	1,00	1,00	1,00
Agra	1,00	0,00	-	1,00	1,00	1,00	0,00	0,00	1,00	1,00	0,00	1,00	1,00	1,00
Amsterdam	0,00	1,00	1,00	-	0,00	1,00	1,00	1,00	0,00	0,00	1,00	0,00	0,00	0,00
Antwerp	0,00	1,00	1,00	0,00	-	1,00	1,00	1,00	0,00	0,00	1,00	0,00	0,00	0,00
Atlanta	1,00	1,00	1,00	1,00	1,00	-	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
Bahrain	1,00	0,00	0,00	1,00	1,00	1,00	-	0,00	1,00	1,00	0,00	1,00	1,00	1,00
Bangkok	1,00	0,00	0,00	1,00	1,00	1,00	0,00	-	1,00	1,00	0,00	1,00	1,00	1,00
Barcelona	0,00	1,00	1,00	0,00	0,00	1,00	1,00	1,00	-	0,00	1,00	0,00	0,00	0,00

Table 8: Hamming distance between some cities according to the attribute Climate



	Aberdeen	Abu_Dhab	Agra	Amsterda	Antwerp	Atlanta	Bahrain	Bangkok	Barcelona	Bath,_Som	Beijing	Benidorm	Berlin	Bilbao
Aberdeen	-	1,00	1,00	0,00	0,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	0,00	0,00
Abu_Dhabi	1,00	-	1,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
Agra	1,00	1,00	-	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
Amsterdam	0,00	1,00	1,00	-	0,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	0,00	0,00
Antwerp	0,00	1,00	1,00	0,00	-	1,00	1,00	1,00	1,00	0,00	1,00	1,00	0,00	0,00
Atlanta	1,00	1,00	1,00	1,00	1,00	-	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
Bahrain	1,00	0,00	1,00	1,00	1,00	1,00	-	1,00	1,00	1,00	1,00	1,00	1,00	1,00
Bangkok	1,00	1,00	1,00	1,00	1,00	1,00	1,00	-	1,00	1,00	1,00	1,00	1,00	1,00
Barcelona	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	-	1,00	1,00	0,00	1,00	1,00

3.1.3 Semantic attributes

Finally, the third component of the compatibility distance measure corresponds to the contribution of the semantic features. Semantic similarity quantifies how words extracted from documents or textual descriptions are alike. There are several exiting measures to compute the semantic similarity, which are based on the knowledge provided by the domain ontologies. In this case we will use the DAMASK ontology.

Superconcept-based distance (SCD) (Batet et al., 2010) has been selected after the analysis of its behaviour in different datasets. The article that presents the method purposes the amount of non-shared superconcepts (ascendant is-a relations) in an ontology as an indication of distance. However, also takes into account the number of shared information between two concepts in order to distinguish concepts that are more specific terms that share more is-a relations in the taxonomy.

The aim the SCD is to present an alternative semantic similarity algorithm, because the classical approaches like *Wu & Palmer* (Wu et al., 1994) and *Leacock & Chodorow* (Leacock et al., 1998) rely on WordNet as the ontology to obtain the similarities between terms. However, WordNet's coverage of some fields like the biomedical domain is limited, and the performance obtained when comparing specific domain concepts is poor.

The SCD definition for comparing a pair of concepts c_i and c_j is based on the following premises:

- Let us define the full concept hierarchy or taxonomy (H^C) of concepts (C) of an ontology as a transitive is-a relation $H^C \in C \times C$.
- Let us define the set $\mathcal{A}(c_i)$ that contains the concept c_i and all the superconcepts (*i.e.*, ancestors) of c_i in a given taxonomy as:

$$\mathcal{A}(c_i)=\{c_j \in C \mid c_j \text{ is superconcept of } c_i \} \cup \{ c_i \} \tag{8}$$

Then the Euclidean-based SuperConcept-based distance (SCD) is defined as

$$SCD(c_i, c_j) = \sqrt{\frac{|\mathcal{A}(c_i) \cup \mathcal{A}(c_j)| - |\mathcal{A}(c_i) \cap \mathcal{A}(c_j)|}{|\mathcal{A}(c_i) \cup \mathcal{A}(c_j)|}} \tag{9}$$

This is the squared root of the number of different ancestors divided by the number of total ancestors (the union). Let's calculate for instance the distance between *Church* and *Mosque* in the ontology represented in Figure 17.

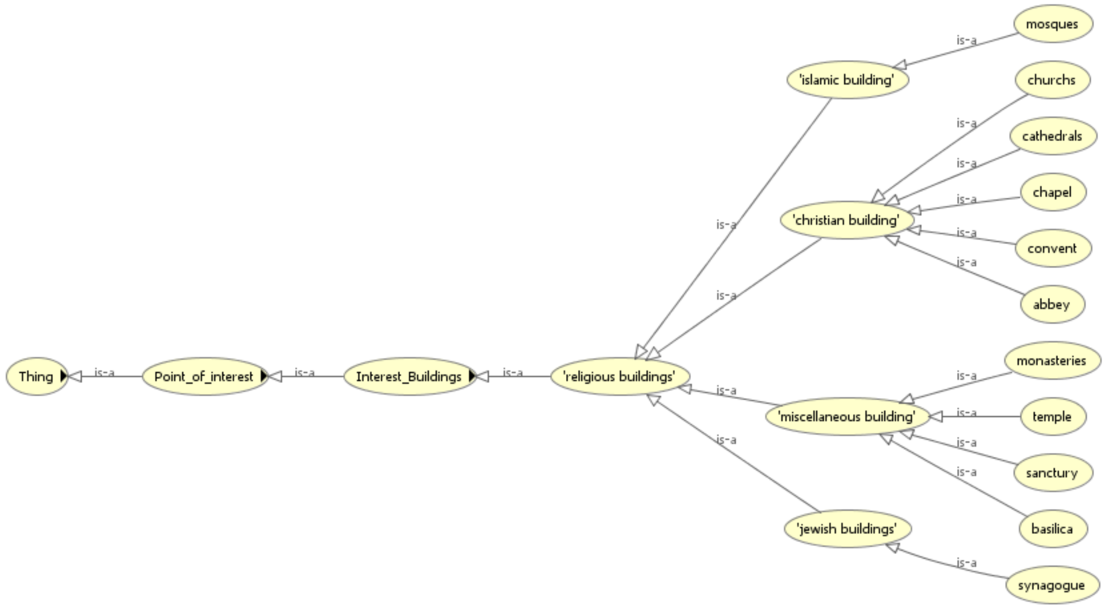


Figure 17: DAMASK ontology portion related to religious buildings

Church and mosque have 4 different ancestors (SCD requires counting itself as ancestor) and the union of ancestors is 8. Hence, the distance between Church and mosque is $\sqrt{4/8}$ = 0,7. The squared root is used to smooth the result and highlight the small differences.

3.1.3.1 An extension of SCD to be used in multi-valued attributes

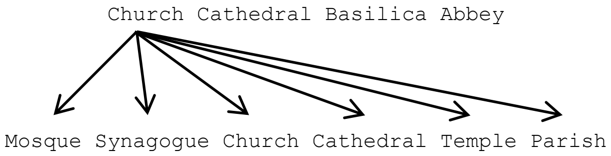
In the CITIES data matrix the attributes corresponding to semantic features usually have more than one value (see the chapter 2). Therefore, the case of multi-valued semantic attributes has been studied. Let us consider an example: calculate the distance between Barcelona and Berlin regarding their religious buildings. The values of these cities are the following. According to the elicitation process we know that all the values correspond to some concept in the DAMASK ontology (so we do not require considering the case of values that are not found in the ontology).

Barcelona	Church#Cathedral#Basilica#Abbey
Berlin	Mosque#Synagogue#Church#Cathedral#Temple#Parish

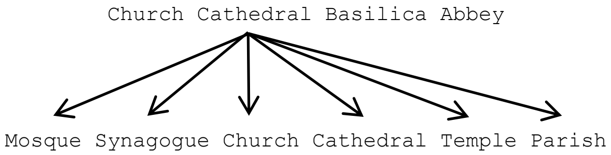
The algorithm used to determine the distance between two cities is as follows:

1. Take a concept value of city A and calculate its semantic distance to each concept of city B, using the ontology. This results in an array of distances.
2. Take the minimum distance on this array and register it in an auxiliary array. In the example above, the minimum distance of Church (in Barcelona) to all the values in Berlin is 0.0 (Church also in Berlin).
3. Repeat this process for all the concepts in A with respect to B.
4. Repeat this process for all the concepts in B with respect to A.
5. Aggregate all the partial distance values with the Ordered Weighted Average (OWA) operator.

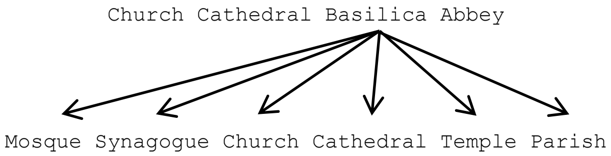
Let us consider the example of comparing Barcelona and Berlin. The first step makes the following calculations:


Church Cathedral Basilica Abbey
Mosque Synagogue Church Cathedral Temple Parish

This will result in an array of distances like these: 0.7, 0.7, 0.0, 0.2, 0.7 and 0.9. The minimum is 0.0.

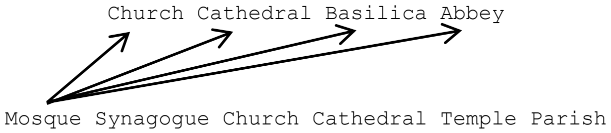

Church Cathedral Basilica Abbey
Mosque Synagogue Church Cathedral Temple Parish

As before, both cities have a cathedral, so the distance here to save is also 0.0. Now in the array we have [0.0, 0.0].


Church Cathedral Basilica Abbey
Mosque Synagogue Church Cathedral Temple Parish

Now, this results in an array of distances like these: 0.7, 0.7, 0.7, 0.7, 0.2 and 0.9. The minimum distance now is between Basilica and Temple (0.2); we save it in the array -> [0.0, 0.0, 0.2].

The process continues for all the concepts for city A, and then we do the same from city B to city A:


Church Cathedral Basilica Abbey
Mosque Synagogue Church Cathedral Temple Parish

When all the distances are calculated, we will end with an array like this: [0.0, 0.0, 0.2, 0.2, 0.7, 0.7, 0.0, 0.0, 0.2, 0.9]. Next is to apply the operator OWA to this array.

This process can be summarised in the following definition.

Definition 1: SuperConcept-based distance for multi-valued attributes (SCD_{mv})

$$SCD_{mv}(i,i') = OWA_{\omega}\left(\left\{\forall c_i : min_{\forall c_{i'}}\left(SCD(c_i, c_{i'})\right)\right\} \cup \left\{\forall c_{i'} : min_{\forall c_i}\left(SCD(c_i, c_{i'})\right)\right\}\right) \tag{10}$$

Finally, the distance for semantic attributes that is used in the compatibility measure is a weighted average of the SuperConcept distances obtained for each of the attributes:

$$d_k^S(i,i') = \sum_{k=1}^{|S|} w_k \, SCD_{mv}(i,i') \tag{11}$$

This approach to multi-valued data is based on the aggregation operation OWA. This operator was defined by R.R. Yager in (Yager, 1988). Since its appearance, it has been studied by many authors and it has been widely applied to many decision making problems (Herrera et al., 1996; Merigo et al., 2009; Xu, 2006; Beliakov et al., 2007).

Definition 2: A function $F: R^n \rightarrow R$ is an OWA operator of dimension n if it has an associated vector ω of dimension n such that its components satisfy:

a. $\omega_j \, \epsilon \, [0,1]$
b. $\sum_{j=1}^{n} \omega_j = 1$

And:

$$F(a_1, a_2, \dots, a_n) = \sum_{j=1}^{n} \omega_j b_j \tag{12}$$

, where b_j is the *j-th* largest element of the bag $\langle a_1, a_2, \dots, a_n \rangle$.

Notice that the fundamental aspect of this operator is the re-ordering step, in particular an argument a_i is not associated with a particular weight w_i but rather a weight is associated with a particular ordered position of argument.

The set of weights is extremely important in the OWA method, because it determines the aggregation policy that the decision maker is imposing on the decision process. Some measures have been introduced to characterize a weight vector, such as evaluating its **attitudinal-character** (or **orness**), which is defined as (Yager, 1988):

$$\alpha(\omega) = \frac{1}{n-1} \sum_{i=0}^{n} \omega_i \, (n-i) \tag{13}$$

It is known that $\alpha \, \epsilon \, [0,1]$. As a general rule, as the allocation of weight in W moves to the top, then α gets closer to one, meanwhile as the weights move to the bottom, α gets closer to zero. Furthermore, if W is symmetrical, then $\alpha(\omega) = 0.5$. This measure provides a characterization of the type of aggregation being performed. An α value near one indicates a bias toward considering mainly the larger values in the argument (i.e. high **orness** or disjunctive behaviour), while an α value near zero indicates preference is being given to the smaller values in the argument (i.e. high **andness** or conjunctive behaviour). An α value near 0.5 is an indication of a neutral type aggregation (i.e. averaging).

3.1.3.2 Generation of the OWA weights

When comparing pairs of cities, the number of arguments (i.e. partial distances) to aggregate is not a constant, it depends on the number of concepts that each city has, consequently we cannot use predetermined OWA weights. For example, if one city has 2 religious buildings and it is compared with another city with 5 religions buildings, with the process described in xx, we will generate 2 × 5 = 10 similarity values to be aggregated using the OWA operator.

In this section, three different methods for generating automatically the set of OWA weights are analysed.

1. **Borda-Kendall law:** which uses a linear function (Lamata et al., 2009; Lamata et al., 2012):

$$\omega_i = \frac{2(n+1-i)}{n(n+1)} \tag{14}$$

The resulting array of weights for *n=10* is: [0.182, 0.164, 0.145, 0.127, 0.109, 0.091, 0.073, 0.055, 0.036, 0.018]. This is the graphic of weights:

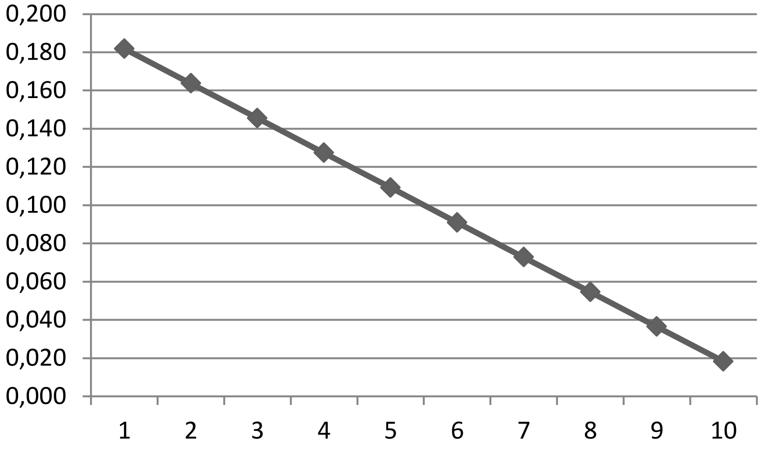


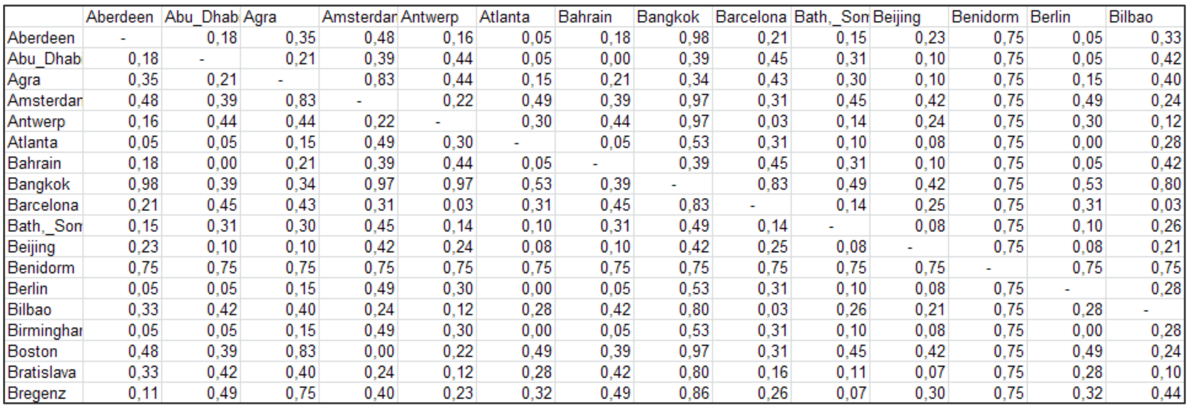
Figure 18: Graphic of the resulting OWA weights using a linear function

If we apply these weights to the ordered (from lower to higher) array of distances and sum them all, we obtain the distance between the two cities. Following the previous example, for the *religious buildings* attribute, the distance between Barcelona and Berlin: 0.134

Notice that this method gives more weight to the similarities than to the differences. It has high *orness* with a value of 0.67.

This process is executed by a Java application that will result in various excel files (Table 9), one per each column in the main data matrix, containing all the distances between each of the cities.

Table 9: Example of the result for the “Religious buildings” attribute



	Aberdeen	Abu_Dhab	Agra	Amsterdar	Antwerp	Atlanta	Bahrain	Bangkok	Barcelona	Bath,_Son	Beijing	Benidorm	Berlin	Bilbao
Aberdeen	-	0,18	0,35	0,48	0,16	0,05	0,18	0,98	0,21	0,15	0,23	0,75	0,05	0,33
Abu_Dhab	0,18	-	0,21	0,39	0,44	0,05	0,00	0,39	0,45	0,31	0,10	0,75	0,05	0,42
Agra	0,35	0,21	-	0,83	0,44	0,15	0,21	0,34	0,43	0,30	0,10	0,75	0,15	0,40
Amsterdar	0,48	0,39	0,83	-	0,22	0,49	0,39	0,97	0,31	0,45	0,42	0,75	0,49	0,24
Antwerp	0,16	0,44	0,44	0,22	-	0,30	0,44	0,97	0,03	0,14	0,24	0,75	0,30	0,12
Atlanta	0,05	0,05	0,15	0,49	0,30	-	0,05	0,53	0,31	0,10	0,08	0,75	0,00	0,28
Bahrain	0,18	0,00	0,21	0,39	0,44	0,05	-	0,39	0,45	0,31	0,10	0,75	0,05	0,42
Bangkok	0,98	0,39	0,34	0,97	0,97	0,53	0,39	-	0,83	0,49	0,42	0,75	0,53	0,80
Barcelona	0,21	0,45	0,43	0,31	0,03	0,31	0,45	0,83	-	0,14	0,25	0,75	0,31	0,03
Bath,_Son	0,15	0,31	0,30	0,45	0,14	0,10	0,31	0,49	0,14	-	0,08	0,75	0,10	0,26
Beijing	0,23	0,10	0,10	0,42	0,24	0,08	0,10	0,42	0,25	0,08	-	0,75	0,08	0,21
Benidorm	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	-	0,75	0,75
Berlin	0,05	0,05	0,15	0,49	0,30	0,00	0,05	0,53	0,31	0,10	0,08	0,75	-	0,28
Bilbao	0,33	0,42	0,40	0,24	0,12	0,28	0,42	0,80	0,03	0,26	0,21	0,75	0,28	-
Birminghar	0,05	0,05	0,15	0,49	0,30	0,00	0,05	0,53	0,31	0,10	0,08	0,75	0,00	0,28
Boston	0,48	0,39	0,83	0,00	0,22	0,49	0,39	0,97	0,31	0,45	0,42	0,75	0,49	0,24
Bratislava	0,33	0,42	0,40	0,24	0,12	0,28	0,42	0,80	0,16	0,11	0,07	0,75	0,28	0,10
Bregenz	0,11	0,49	0,75	0,40	0,23	0,32	0,49	0,86	0,26	0,07	0,30	0,75	0,32	0,44

2. **Non-linear decreasing function**. We have defined a function that generates a set of weights in a non-linear decreasing way as shown in Figure 19. The equation used is the following:

$$\omega_i = \frac{1}{x^{\frac{4}{5}}} \tag{15}$$

This will result in an array of weights that do not sum 1. To solve this, all the resulting values are summed and then each one is divided by this resulting sum, like normalization. Now the weights sum 1 as expected, and this are their values for *n=10*: [0.281 0.161 0.116 0.093 0.077 0.067 0.059 0.053 0.048 0.044]. Its *orness* is 0.69 and this is the graphic of weights:

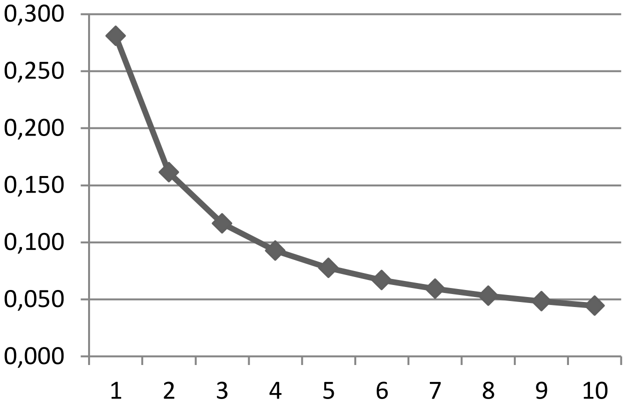
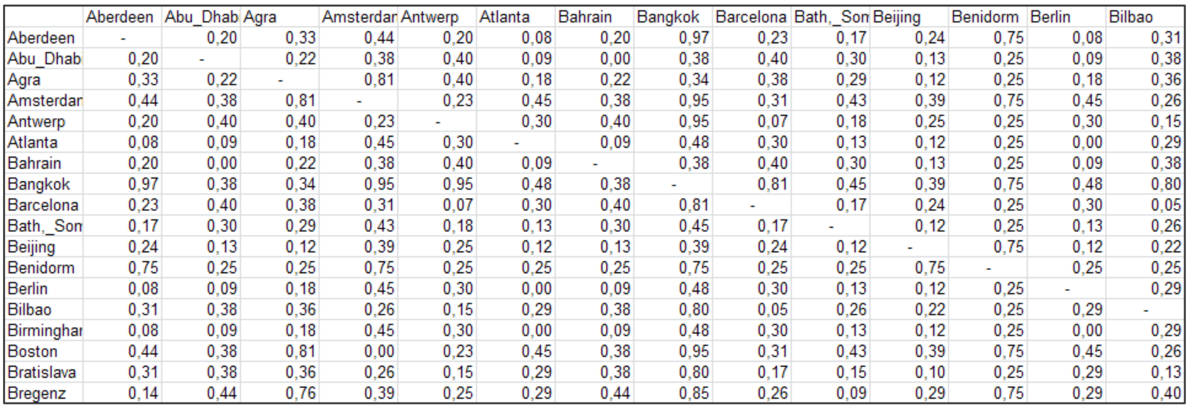


Figure 19: Graphic of the resulting weights using a non-linear decreasing function

Table 10: Example of the result for the “Religious buildings” attribute using the non-linear decreasing set of weights for OWA



	Aberdeen	Abu_Dhab	Agra	Amsterdar	Antwerp	Atlanta	Bahrain	Bangkok	Barcelona	Bath,_Son	Beijing	Benidorm	Berlin	Bilbao
Aberdeen	-	0,20	0,33	0,44	0,20	0,08	0,20	0,97	0,23	0,17	0,24	0,75	0,08	0,31
Abu_Dhab	0,20	-	0,22	0,38	0,40	0,09	0,00	0,38	0,40	0,30	0,13	0,25	0,09	0,38
Agra	0,33	0,22	-	0,81	0,40	0,18	0,22	0,34	0,38	0,29	0,12	0,25	0,18	0,36
Amsterdar	0,44	0,38	0,81	-	0,23	0,45	0,38	0,95	0,31	0,43	0,39	0,75	0,45	0,26
Antwerp	0,20	0,40	0,40	0,23	-	0,30	0,40	0,95	0,07	0,18	0,25	0,25	0,30	0,15
Atlanta	0,08	0,09	0,18	0,45	0,30	-	0,09	0,48	0,30	0,13	0,12	0,25	0,00	0,29
Bahrain	0,20	0,00	0,22	0,38	0,40	0,09	-	0,38	0,40	0,30	0,13	0,25	0,09	0,38
Bangkok	0,97	0,38	0,34	0,95	0,95	0,48	0,38	-	0,81	0,45	0,39	0,75	0,48	0,80
Barcelona	0,23	0,40	0,38	0,31	0,07	0,30	0,40	0,81	-	0,17	0,24	0,25	0,30	0,05
Bath,_Son	0,17	0,30	0,29	0,43	0,18	0,13	0,30	0,45	0,17	-	0,12	0,25	0,13	0,26
Beijing	0,24	0,13	0,12	0,39	0,25	0,12	0,13	0,39	0,24	0,12	-	0,75	0,12	0,22
Benidorm	0,75	0,25	0,25	0,75	0,25	0,25	0,25	0,75	0,25	0,25	0,75	-	0,25	0,25
Berlin	0,08	0,09	0,18	0,45	0,30	0,00	0,09	0,48	0,30	0,13	0,12	0,25	-	0,29
Bilbao	0,31	0,38	0,36	0,26	0,15	0,29	0,38	0,80	0,05	0,26	0,22	0,25	0,29	-
Birminghar	0,08	0,09	0,18	0,45	0,30	0,00	0,09	0,48	0,30	0,13	0,12	0,25	0,00	0,29
Boston	0,44	0,38	0,81	0,00	0,23	0,45	0,38	0,95	0,31	0,43	0,39	0,75	0,45	0,26
Bratislava	0,31	0,38	0,36	0,26	0,15	0,29	0,38	0,80	0,17	0,15	0,10	0,25	0,29	0,13
Bregenz	0,14	0,44	0,76	0,39	0,25	0,29	0,44	0,85	0,26	0,09	0,29	0,75	0,29	0,40

These results are similar to the ones seen with the previous method, which only small changes. This is due to the bigger weight given to the first value and to the major dissimilarities.

3. **Linguistic quantifiers:** The classical logic uses just two quantifiers, which are: the universal quantifier ∀ (all) and the existential quantifier ∃ (exists). But one may want to use something in between like *most*, *many*, *at least half*, *some*, and *few*. These are the linguistic quantifiers defined for fuzzy sets in (Yager, 1996; Yager, 1993). They permit to model different compensation aggregation mechanism, applied to define different behavioural policies (from pessimistic – conjunctive – to optimistic – disjunctive).

   After a careful study of the behavioural character of the different approaches, we have decided to use the linguistic quantifier *most* for aggregating the partial distances obtained for semantic attributes. This quantifier models a partial conjunctive policy. It means that a city will be similar to another one if *most* of their values in the semantic attributes are similar. Not permitting the compensation of low values with high ones.

   The weights associated to linguistic quantifiers are usually obtained from Fuzzy Quantifiers (Yager, 1996). A function Q : [0, 1] → [0, 1] *is a* regular monotonically non-decreasing fuzzy quantifier *(non-decreasing fuzzy quantifiers for short)* if it satisfies:

   i. $Q(0) = 0$;
   ii. $Q(1) = 1$;
   iii. $x > y$ implies $Q(x) \geq Q(y)$.

   A well-known fuzzy quantifier is based on the sigmoidal function and it is given by the following definition.

$$Q^{\alpha}(x) = f(x) = \begin{cases} 0, & if\ x = 0 \\ \frac{1}{1 + e^{(\alpha - x)*10}} \text{ for } \alpha > 0, & if\ 0 < x < 1 \\ 1, & if\ x = 1 \end{cases} \qquad (15)$$

   A graphical representation of this fuzzy quantifier is given in Figure 20 for some particular values on the parameter α, α = {0, 0.1, . . . 0.9}. We can observe that for small a values, the function increases quickly near x = 0,whereas the increase is smoothly for larger values of α.

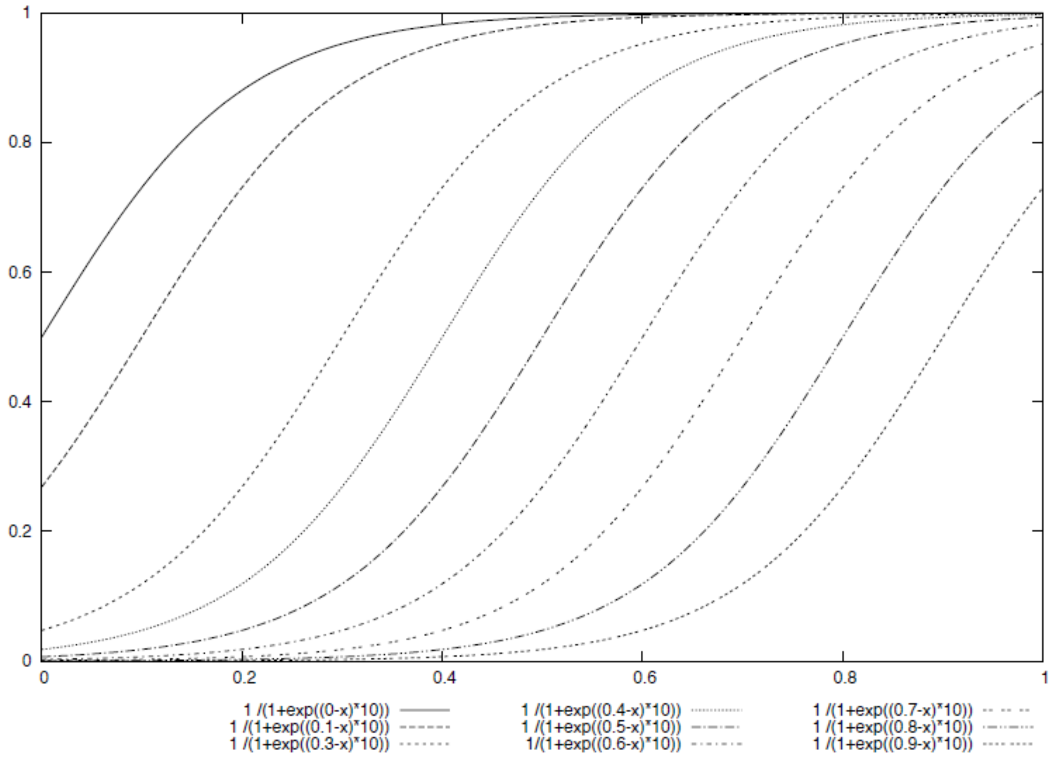


Figure 20: Representation of function 15 for α 0 to 0.9.

Using this fuzzy quantifier, the OWA weights can be obtained with the following equation:

$$w_i = \left[Q\left(\frac{i}{N}\right) - Q\left(\frac{i-1}{N}\right)\right] \tag{16}$$

For the linguistic quantifier "most", the recommended value is α = 0.6. Using Eq 16 and Eq 15, the set of weights obtained to aggregate 10 values is shown in Figure 21. Taking into account that the OWA operator will sort the values in a decreasing way, we are giving high weights for those values that are below the median, which assures that all the previous ones are equal or higher. Different values of the parameter α would shift the curve in the figure 21 to the left for lower values and to the right otherwise.

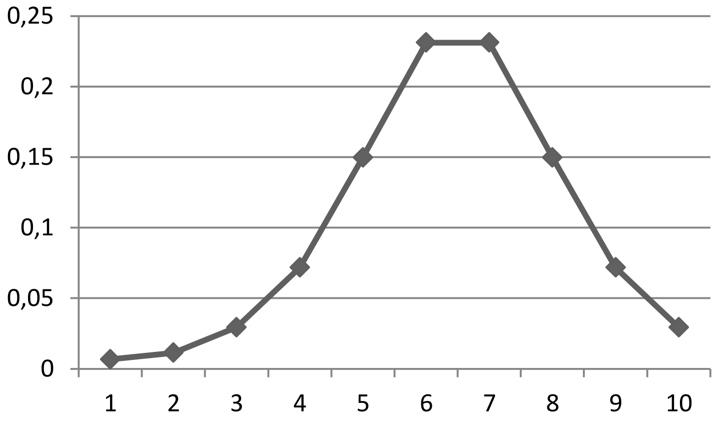


Figure 21: Graphic of the resulting weights using the linguistic quantifier defined by Eq. 16 with α = 0.6 for n = 10

The results obtained with this set of weights in the OWA operator are quite different from the ones resulting when applying the previous methods: for *n=10* the resulting weights are [0.007 0.011 0.029 0.072

0.150 0.231 0.231 0.150 0.072 0.029] and its *orness* is 0.39. In this case, the weight for the most similar and the most dissimilar concepts of the array are low, so that if most of the cities do only coincide in 1 value (f.i. in religious buildings, almost every city has a church), the result is a high value of similarity using the methods 1 and 2, but not with this one.

Here is another example represented in Figure 22 for *n=4*, which results in weights [0.029 0.239 0.548 0.164] and an *orness* of 0.28.

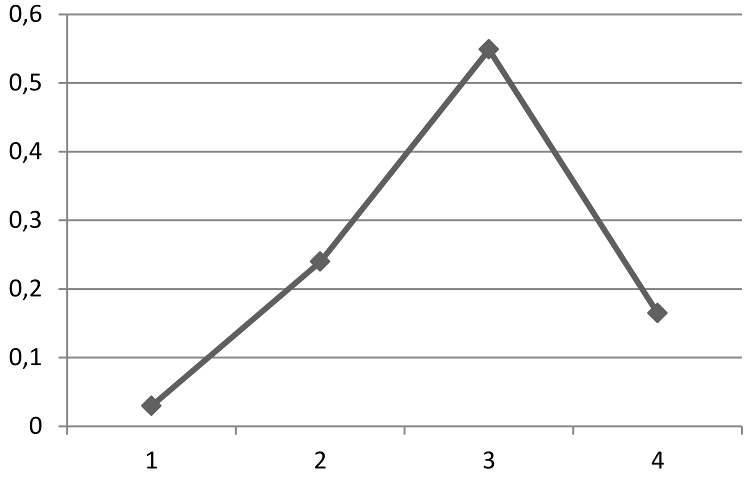
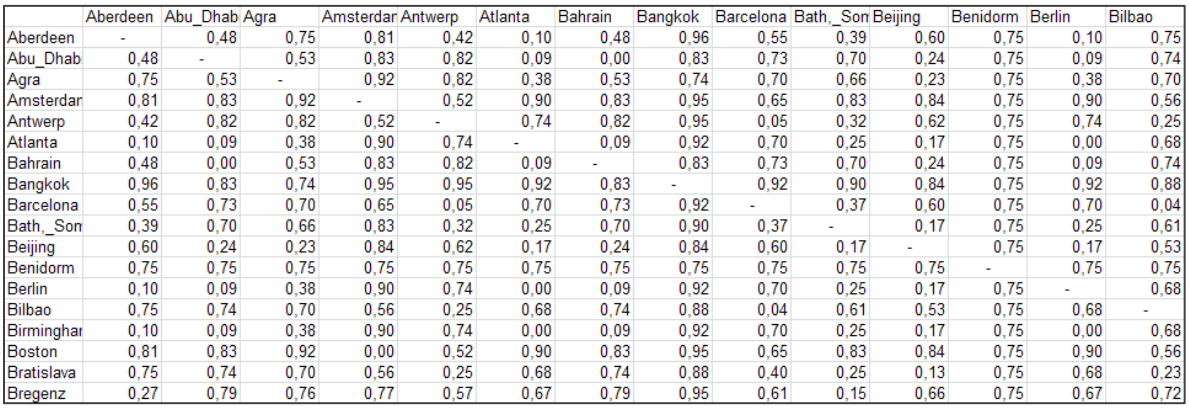


Figure 22: Graphic of the resulting weights using the linguistic quantifier defined by Eq. 16 with α = 0.6 for n = 4

It can be seen that as explained before, the most similar and the most dissimilar values have low weights. The orness values obtained indicate that in this case we are taking a more conjunctive behaviour, less compensative. With this approach we are able to stress the differences, enhancing the discriminating power of semantic features. This is an interesting result for clustering purposes, so this third approach is the one that will be included in the recommender system.

Table 11: Example of the results for the attribute “Religious buildings” using the linguistic quantifier “most”.



	Aberdeen	Abu_Dhab	Agra	Amsterdar	Antwerp	Atlanta	Bahrain	Bangkok	Barcelona	Bath,_Son	Beijing	Benidorm	Berlin	Bilbao
Aberdeen	-	0,48	0,75	0,81	0,42	0,10	0,48	0,96	0,55	0,39	0,60	0,75	0,10	0,75
Abu_Dhab	0,48	-	0,53	0,83	0,82	0,09	0,00	0,83	0,73	0,70	0,24	0,75	0,09	0,74
Agra	0,75	0,53	-	0,92	0,82	0,38	0,53	0,74	0,70	0,66	0,23	0,75	0,38	0,70
Amsterdar	0,81	0,83	0,92	-	0,52	0,90	0,83	0,95	0,65	0,83	0,84	0,75	0,90	0,56
Antwerp	0,42	0,82	0,82	0,52	-	0,74	0,82	0,95	0,05	0,32	0,62	0,75	0,74	0,25
Atlanta	0,10	0,09	0,38	0,90	0,74	-	0,09	0,92	0,70	0,25	0,17	0,75	0,00	0,68
Bahrain	0,48	0,00	0,53	0,83	0,82	0,09	-	0,83	0,73	0,70	0,24	0,75	0,09	0,74
Bangkok	0,96	0,83	0,74	0,95	0,95	0,92	0,83	-	0,92	0,90	0,84	0,75	0,92	0,88
Barcelona	0,55	0,73	0,70	0,65	0,05	0,70	0,73	0,92	-	0,37	0,60	0,75	0,70	0,04
Bath,_Son	0,39	0,70	0,66	0,83	0,32	0,25	0,70	0,90	0,37	-	0,17	0,75	0,25	0,61
Beijing	0,60	0,24	0,23	0,84	0,62	0,17	0,24	0,84	0,60	0,17	-	0,75	0,17	0,53
Benidorm	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	-	0,75	0,75
Berlin	0,10	0,09	0,38	0,90	0,74	0,00	0,09	0,92	0,70	0,25	0,17	0,75	-	0,68
Bilbao	0,75	0,74	0,70	0,56	0,25	0,68	0,74	0,88	0,04	0,61	0,53	0,75	0,68	-
Birminghar	0,10	0,09	0,38	0,90	0,74	0,00	0,09	0,92	0,70	0,25	0,17	0,75	0,00	0,68
Boston	0,81	0,83	0,92	0,00	0,52	0,90	0,83	0,95	0,65	0,83	0,84	0,75	0,90	0,56
Bratislava	0,75	0,74	0,70	0,56	0,25	0,68	0,74	0,88	0,40	0,25	0,13	0,75	0,68	0,23
Bregenz	0,27	0,79	0,76	0,77	0,57	0,67	0,79	0,95	0,61	0,15	0,66	0,75	0,67	0,72

3.1.3.2.1 Comparison of the different methods for generating OWA weights

In order to apreciate the differences between the three different techniques for generating automatically the set of weights, a comparative has been done in table 12.

Table 12: Comparative of the 3 methods for the “Religious buildings” attribute.

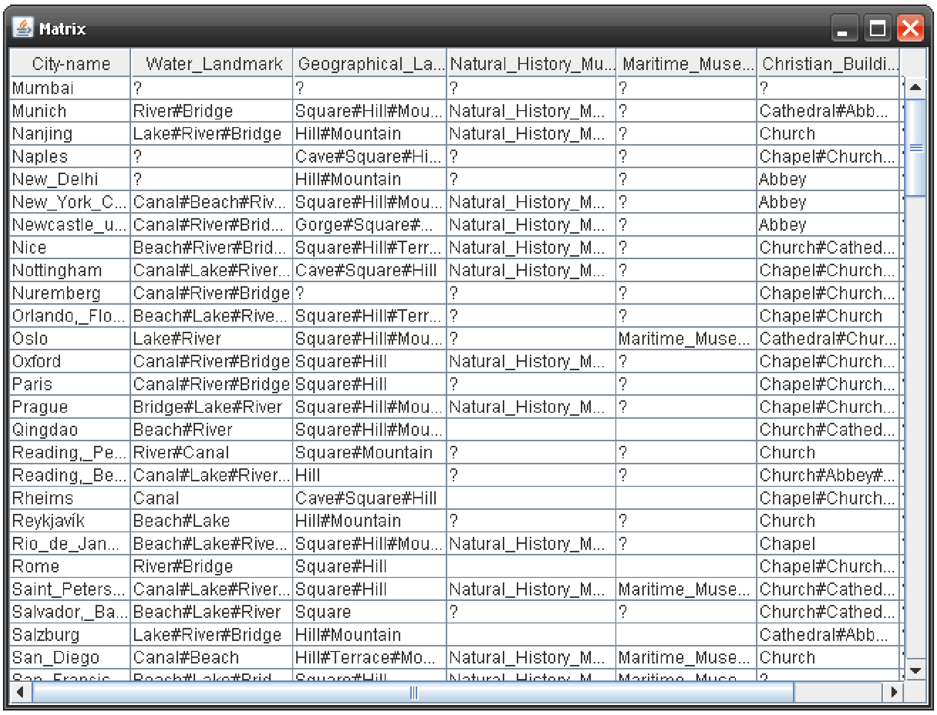
	Linear OWA	Non-linear OWA	Linguistic qualifier OWA
Aberdeen – Agra	0.35	0.33	0.75
Antwerp – Barcelona	0.03	0.07	0.05
Agra – Amsterdam	0.83	0.81	0.92
Aberdeen – Bangkok	0.98	0.97	0.96
Barcelona – Budapest	0.11	0.15	0.26

It is easy to see that the distances are a bit larger if the linguistic qualifier *most* is used, because it is more pessimistic than the two first approaches. This specially noted between Aberdeen and Agra, because we have decreased the compensation factor. For cities with a lot of values in common, such as Antwerp and Barcelona, the difference is small. Similarly, with cities with very few things in common, such as Aberdeen and Bangkok, the three approaches give also a quite similar distance value. Further analysis will be done using the results of the clustering.

3.1.4 Treatment of the missing values

The treatment of missing values must deserve special attention in clustering algorithms. In the CITIES data matrix that has been compiled in the DAMASK project we only find missing values in the semantic attributes. The numerical and categorical information is complete for all the cities, because we have used different extraction mechanisms until obtaining all the data, as explained in chapter 2.

For the case of semantic descriptions, in case that the system is not able to find any evidence for a given attribute, the symbol '?' is used. As explained in chapter 2, in this case, the symbol is not exactly representing a missing value as normally understood, because the lack of information about one attribute is telling us that probably the city does not have any instance of this type. For example, in Figure 23, the automatic extraction system has not found any information about Maritime Museums in Munich, because certainly they do not exist. The data matrix construction by means of extraction processes is slanted by the *precision* and *recall* of each method used during the whole process (i.e., the natural language parser, the named entity detection heuristics, the inaccuracy of Web statistics and the relat-edness measures). The *precision index* measures the number of correct values among all the values obtained. *Recall* is calculated by dividing the number of correct values by the total of values that could have been found. As explained in deliverable D2, for the purpose of the project, high precision is needed, to ensure that the values that we attach to some city are correct. High precision is achieven at a cost of reducing the recall. In this case, the symbol '?' may appear in the data matrix because we have not been able to retrieve the information from the Web page. For example Oslo has a Natural History Museum, but this data has not been found by the system.



City-name	Water_Landmark	Geographical_La...	Natural_History_Mu...	Maritime_Muse...	Christian_Buildi...
Mumbai	?	?	?	?	?
Munich	River#Bridge	Square#Hill#Mou...	Natural_History_M...	?	Cathedral#Abb...
Nanjing	Lake#River#Bridge	Hill#Mountain	Natural_History_M...	?	Church
Naples	?	Cave#Square#Hi...	?	?	Chapel#Church...
New_Delhi	?	Hill#Mountain	?	?	Abbey
New_York_C...	Canal#Beach#Riv...	Square#Hill#Mou...	Natural_History_M...	?	Abbey
Newcastle_u...	Canal#River#Brid...	Gorge#Square#...	Natural_History_M...	?	Abbey
Nice	Beach#River#Brid...	Square#Hill#Terr...	Natural_History_M...	?	Church#Cathed...
Nottingham	Canal#Lake#River...	Cave#Square#Hill	Natural_History_M...	?	Chapel#Church...
Nuremberg	Canal#River#Bridge	?	?	?	Chapel#Church...
Orlando,_Flo...	Beach#Lake#Rive...	Square#Hill#Terr...	?	?	Chapel#Church
Oslo	Lake#River	Square#Hill#Mou...	?	Maritime_Muse...	Cathedral#Chur...
Oxford	Canal#River#Bridge	Square#Hill	Natural_History_M...	?	Chapel#Church...
Paris	Canal#River#Bridge	Square#Hill	?	?	Chapel#Church...
Prague	Bridge#Lake#River	Square#Hill#Mou...	Natural_History_M...	?	Chapel#Church...
Qingdao	Beach#River	Square#Hill#Mou...			Church#Cathed...
Reading,_Pe...	River#Canal	Square#Mountain	?	?	Church
Reading,_Be...	Canal#Lake#River...	Hill	?	?	Church#Abbey#...
Rheims	Canal	Cave#Square#Hill			Chapel#Church...
Reykjavik	Beach#Lake	Hill#Mountain	?	?	Church
Rio_de_Jan...	Beach#Lake#Rive...	Square#Hill#Mou...	Natural_History_M...	?	Chapel
Rome	River#Bridge	Square#Hill			Chapel#Church...
Saint_Peters...	Canal#Lake#River...	Square#Hill	Natural_History_M...	Maritime_Muse...	Church#Cathed...
Salvador,_Ba...	Beach#Lake#River	Square	?	?	Church#Cathed...
Salzburg	Lake#River#Bridge	Hill#Mountain			Cathedral#Abb...
San_Diego	Canal#Beach	Hill#Terrace#Mo...	Natural_History_M...	Maritime_Muse...	Church
San_Francis...	Beach#Lake#Brid...	Square#Hill	Natural_History_M...	Maritime_Muse...	?

Figure 23: Resulting matrix from the Tree extracting procedure

The distance for a city that has missing data value to another city with known data has been established to a value of 0.75. A value higher than 0.5 has been fixed in order to represent that a city with something is far from a city with probably no elements of the same typology. Due to the recall error, as a missing cannot absolutly mean that the city has no instances for the attribute, the value of distance used is not 1, but 0.75.

Moreover, the distance between two cities that have missings has been set to 0.25. In this case, this lower value on the distance (i.e. higher similarity) represents that these two cities have something in common as both may be cities without instances on the given attribute. Again, due to the recall error, the distance is set to 0.25 and not 0.

Table 13: Distances applied to semantic values when missing information

	‘?’	**No missing**
‘?’	0.25	0.75
No missing	0.75	Calculated with eq.10

4 Adapting K-means for including multi-type and multi-valued attributes

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 chapter 2 for more details.

In the DAMASK Deliverable D2 (Vicient et al., 2011) 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 chapter 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 (Vicient et al., 2011)).

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.

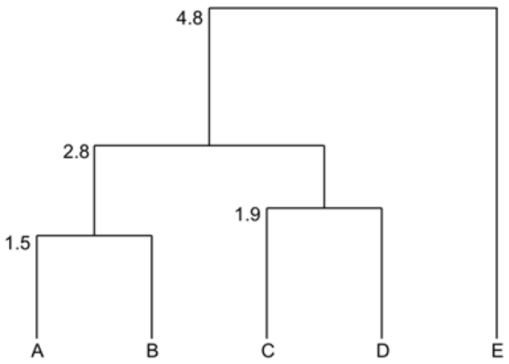


Figure 24: Dendogram

4.1.2 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).

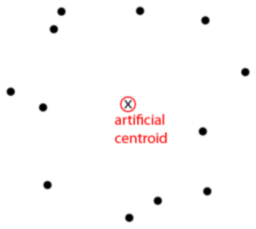


Figure 25: Artificial centroid representation. High precision.

- 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).

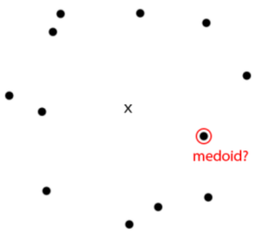


Figure 26: Medoid representation. Low precision.

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.

4.1.3 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 a 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.

4.1.4 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, $\Lambda = [\lambda_{l,i}]$ 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 $v_c \epsilon V$, 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 ϵ [1,+∞)and S is the dimensionality of the vectors.

4.1.5 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 chapter 3.

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.

4.2 Defining a centroid for semantic multi-valued data

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 14). 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 14: 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 term in 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).

4.2.1 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.

4.2.1.1 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.

4.2.1.2 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.

4.2.2 The 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 15.

Table 15: 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 15. 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 16 shows the prototype including the frequency, which is indicated before the concept name.

Table 16: 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

4.2.2.1 Formalization of the centroid

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

$$c = \{\langle n_i, t_i \rangle | n_i > max(n * \lambda, 1)\},$$

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 17: 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.

4.2.2.2 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

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

$$W' = f(W) = \{w'_1, w'_2, \ldots, w'_n\},$$

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.

4.2.2.3 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 27 (called attribute’ length).

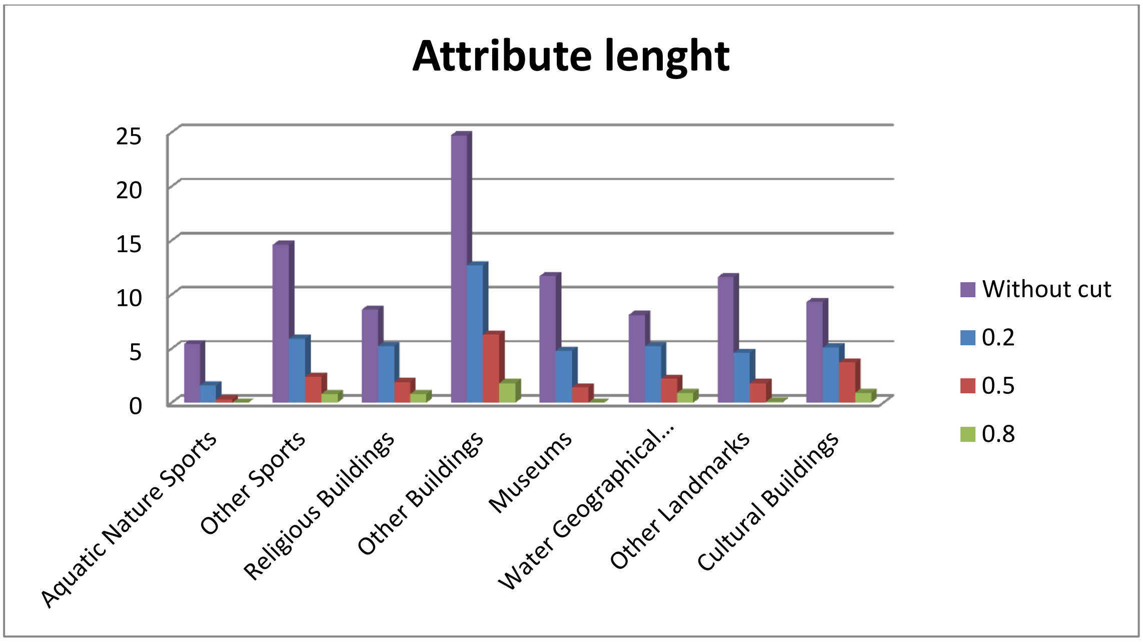


Figure 27: Attribute length after cut for different lamdba values.

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 end up removing too many concepts from the centroid. So that, a threshold of λ = 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.

4.3 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.

4.3.1 The algorithm

The steps of the *k*-means clustering algorithm presented in section 4.1.3 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.

4.3.2 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.

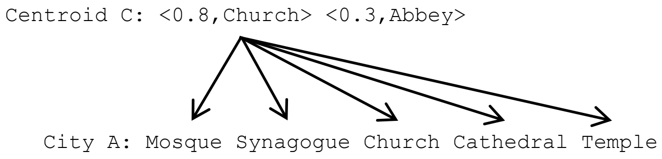
   Each of these cities has its own values for the attributes considered. Two numerical values for Altitude and Population, two categorical values for the attributes Continent and Climate, and then lists of terms associated to each of the 8 semantic attributes (aquatic nature sports, other sports, religious 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 4.2.2., as a set of tuples of the form $\langle n_i, t_i \rangle$. Initially the values of n_i are set to 1, so that all tuples are of the form $\langle 1, t_i \rangle$.

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 (chapter 3). 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:


Centroid C: <0.8,Church> <0.3,Abbey>
City A: Mosque Synagogue Church Cathedral Temple

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 steps 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.

4.4 Implementation

A Java program has been developed to cluster the 150 cities specified in chapter 2. This program follows all the steps presented in section 4.3.

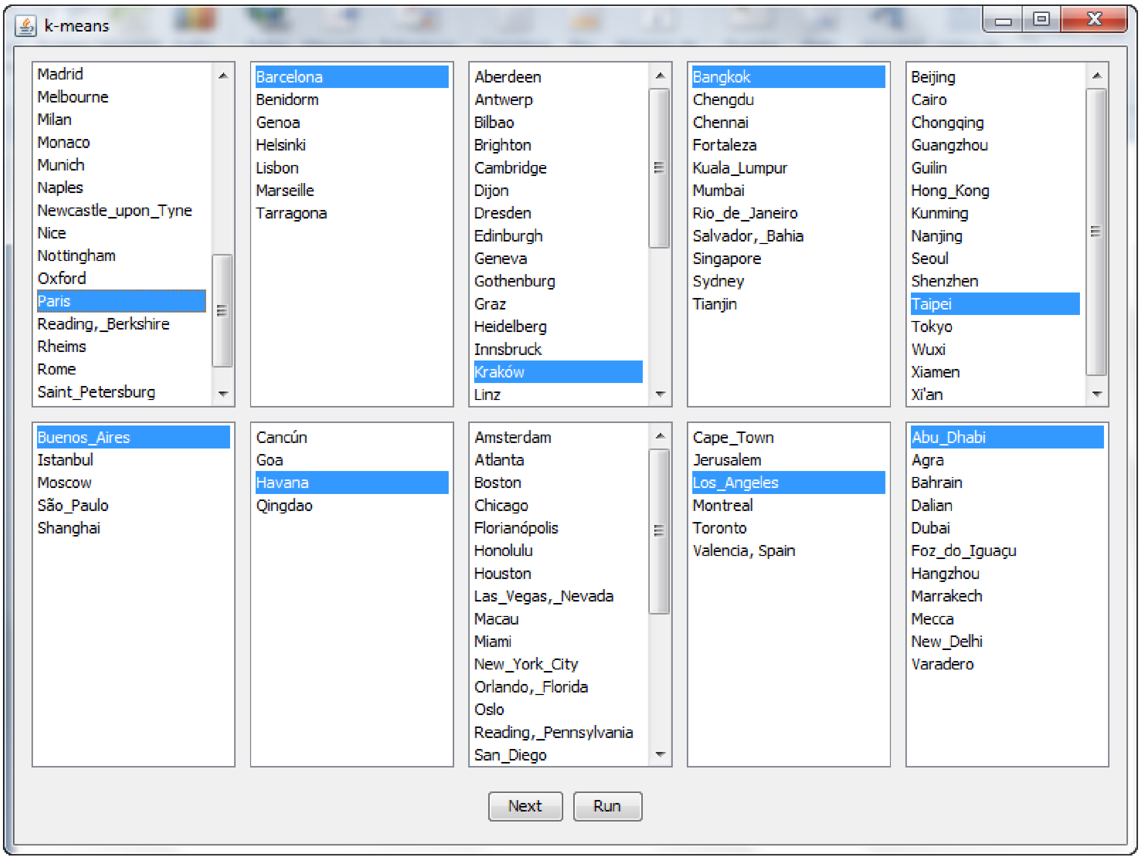
The program has 2 inputs:

- Excel with the absolute distances (numerical + categorical + semantic) between cities pre-calculated using a little program that follows the specification defined in chapter 3.
- An excel file with DAMASK data matrix as defined in chapter 2.

The following parameters have been fixed:

- *k* = 10 for the desired number of clusters.
- λ = 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:



Madrid
Melbourne
Milan
Monaco
Munich
Naples
Newcastle_upon_Tyne
Nice
Nottingham
Oxford
Paris
Reading,_Berkshire
Rheims
Rome
Saint_Petersburg

Barcelona
Benidorm
Genoa
Helsinki
Lisbon
Marseille
Tarragona

Aberdeen
Antwerp
Bilbao
Brighton
Cambridge
Dijon
Dresden
Edinburgh
Geneva
Gothenburg
Graz
Heidelberg
Innsbruck
Kraków
Linz

Bangkok
Chengdu
Chennai
Fortaleza
Kuala_Lumpur
Mumbai
Rio_de_Janeiro
Salvador,_Bahia
Singapore
Sydney
Tianjin

Beijing
Cairo
Chongqing
Guangzhou
Guilin
Hong_Kong
Kunming
Nanjing
Seoul
Shenzhen
Taipei
Tokyo
Wuxi
Xiamen
Xi'an

Buenos_Aires
Istanbul
Moscow
São_Paulo
Shanghai

Cancún
Goa
Havana
Qingdao

Amsterdam
Atlanta
Boston
Chicago
Florianópolis
Honolulu
Houston
Las_Vegas,_Nevada
Macau
Miami
New_York_City
Orlando,_Florida
Oslo
Reading,_Pennsylvania
San_Diego

Cape_Town
Jerusalem
Los_Angeles
Montreal
Toronto
Valencia, Spain

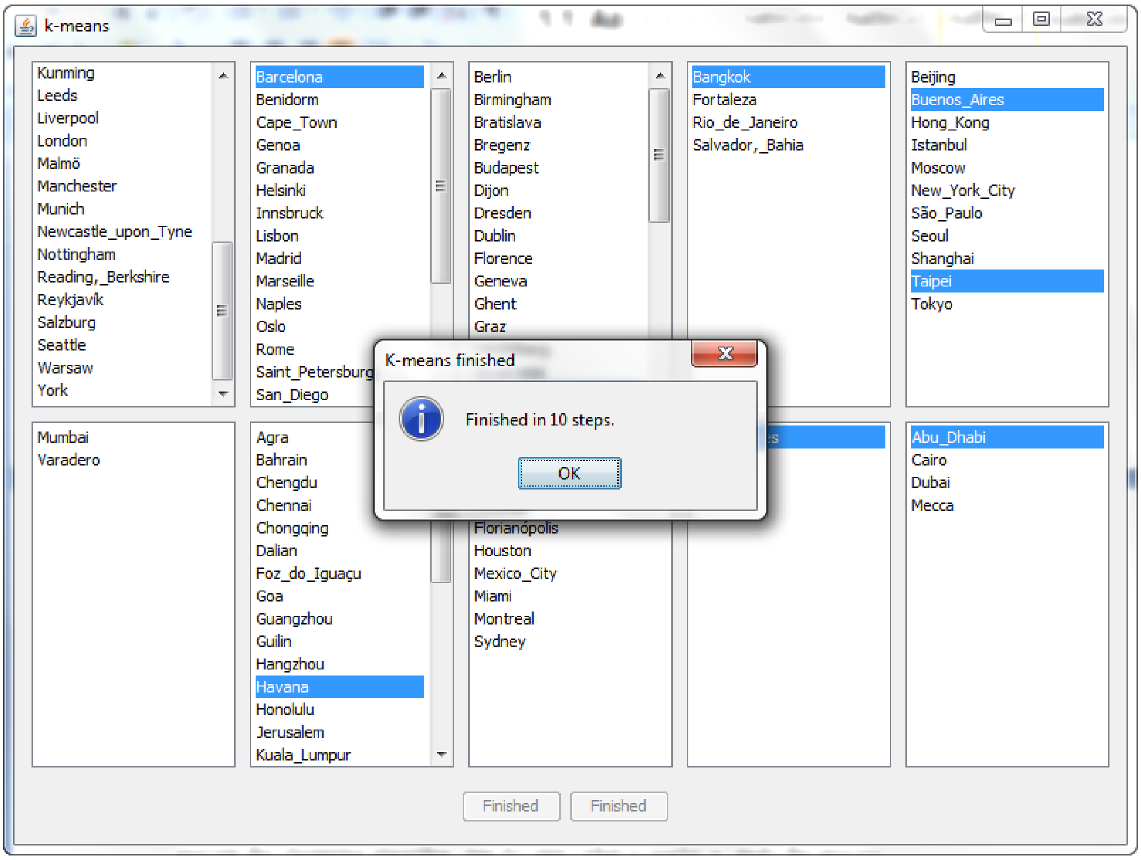
Abu_Dhabi
Agra
Bahrain
Dalian
Dubai
Foz_do_Iguaçu
Hangzhou
Marrakech
Mecca
New_Delhi
Varadero

Figure 28: The results window of the program

In Figure 28, 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.



Kunming
Leeds
Liverpool
London
Malmö
Manchester
Munich
Newcastle_upon_Tyne
Nottingham
Reading,_Berkshire
Reykjavik
Salzburg
Seattle
Warsaw
York

Barcelona
Benidorm
Cape_Town
Genoa
Granada
Helsinki
Innsbruck
Lisbon
Madrid
Marseille
Naples
Oslo
Rome
Saint_Petersburg
San_Diego

Berlin
Birmingham
Bratislava
Bregenz
Budapest
Dijon
Dresden
Dublin
Florence
Geneva
Ghent
Graz

Bangkok
Fortaleza
Rio_de_Janeiro
Salvador,_Bahia

Beijing
Buenos_Aires
Hong_Kong
Istanbul
Moscow
New_York_City
São_Paulo
Seoul
Shanghai
Taipei
Tokyo

Mumbai
Varadero

Agra
Bahrain
Chengdu
Chennai
Chongqing
Dalian
Foz_do_Iguaçu
Goa
Guangzhou
Guilin
Hangzhou
Havana
Honolulu
Jerusalem
Kuala_Lumpur

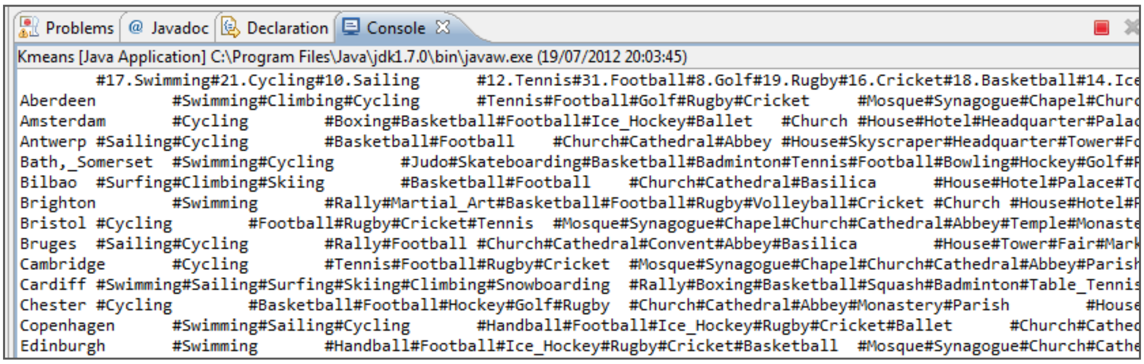
Florianópolis
Houston
Mexico_City
Miami
Montreal
Sydney

Abu_Dhabi
Cairo
Dubai
Mecca

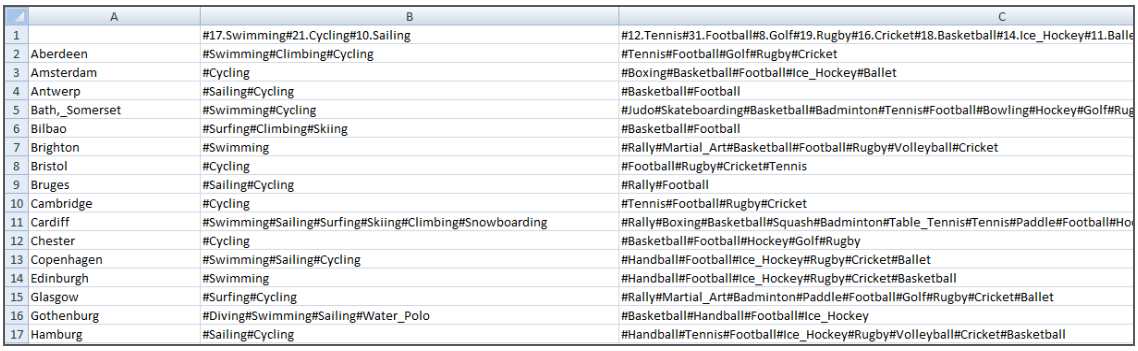
Finished in 10 steps.

Figure 29: 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):



```
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#Fc
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#Tc
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#Cathec
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 30: 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 chapter 6.

5 User-oriented recommender system

This chapter explains the Web-based recommender system developed in the last stage of the DAMASK project. This system is a prototype that will permit to evaluate the performance of the developed methods and techniques in the case study of the field of Tourism.

The system provides a Web page that is connected with a server application to permit to the user to receive a list of touristic destinations that fits with his/her interests and preferences. The systems allows to have multiple users registered, each one linked to a personal profile that stores the preferred values on a set of criteria.

The system is built using the data obtained in previous steps of the project, such as the data matrix explained in chapter 2, and the clusters built using the clustering technique explained in chapter 4.

Briefly, a user defines a set of requirements for the desired type of city using only semantic attributes. The recommender system selects the cluster that is able to fulfill to a highest degree the preferences given by the user. Then, from the cities that belong to this cluster, a selection can be made using some filters defined upon the numerical attributes (population and elevation) and the categorical attributes (continent and climate).

This document is structured in two main sections. The first explains in detail the design of the Web application, the tools used for the implementation and its architecture. The second gives details about the functionalities provided by the application, the interface, as well as the algorithms behind the recommendation process.

5.1 The architecture

The implementation tools have been selected taking into account that Java is the programming language adopted at the beginning of the project. Java facilitates an easy portability to different operating systems and platforms, even to different types of devices.

To deploy the recommender system at the Web, a computer has been configured as Web server. This server has been installed using the tool WAMP (www.wampserver.com), which includes: Windows, Apache, MySQL and PHP. Since the project is just a prototype, without the intention to build a more robust and ad-hoc server to host a system for real exploitation, the WAMP application sufficiently fulfills the needs of the prototype. Apache is used as an HTTP server, MySQL is the database and PHP is no used (since the system is implemented in Java). WAMP also provides a phpMyAdmin to manage the MySQL database. It is worth to note that the Apache server does not work with Java applications per se. For that purpose, Apache Tomcat is used as the main HTTP server to support the development with Java Web applications.

5.1.1 Tools used

5.1.1.1 JAVA + JavaServer Faces

The base of the system is built using Java 7 for an important reason. The previous applications of the DAMASK project were build using JAVA. Hence, the portability of some algorithms to the Web system is much easier if the same language is used. But Java *per se* does not provide tools to program a Web system. For that purpose, the JavaServer Faces technology is used.

JavaServer Faces (JSF) is a Java-based technology or Web application framework designed to simplify and provide tools for the development of Web-based user interfaces. JSF follows the model-view-controller (MVC) model, which is a standard in Web development. Basically, MVC separates the representation of information from the user's interaction with it. The *model* consists of application data and the *controller* is responsible of manage the input to process and command it to the *model* or the *view* (see Figure 31).

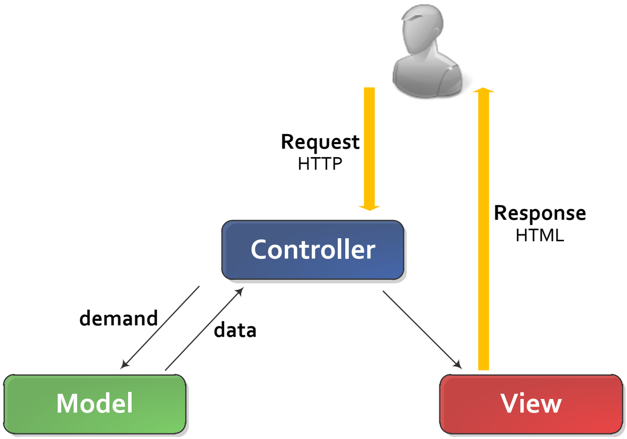


Figure 31: MVC architecture

JavaServer Faces uses JavaServer Pages (JSP) as display technology in its first specification. JSP is a technology used to create dynamically generated Web pages based on HTML, XML, or other document types. But the 2nd specification of the JSF uses Facelets, its own view handler technology. Facelets requires valid input XML documents to work. JSF 2 can also work with JSP, but its use is discouraged.

Facelets provides templating features. This is ideal to build a Web site that reuses at each page the same elements such as the header, the footer, menus… Hence, a master template can be built indicating which source to use as permanent elements of the Web.

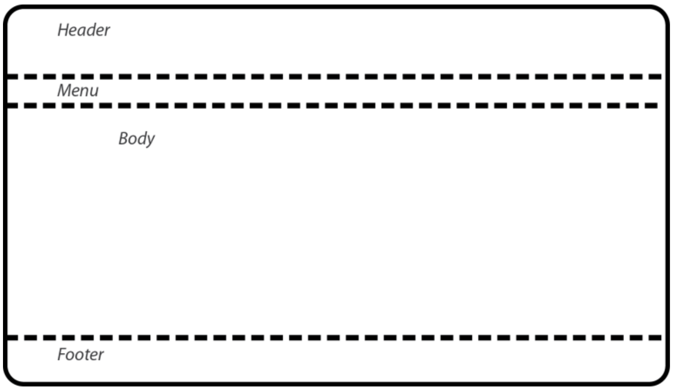


Figure 32: Representation of a master template

When a new page is developed, only the body needs to be written, indicating in the file that the document uses the template *X* and its content replaces the placeholder *Y* ("body", for instance).

5.1.1.2 MySQL + Hibernate

The WAMP server includes a MySQL database. MySQL is a relational database management system (RDBMS) that provides multi-user access to a number of databases. As its name suggests, MySQL uses the Structured Query Language (SQL).

MySQL databases can be managed and administered using phpMyAdmin that WAMP also includes. Some of the phpMyAdmin feature includes the creation, modification and deletion of databases, tables, fields and rows, execute SQL statements, or manage users and its permissions.

It is not trivial to use a MySQL database with Java. There exists some implementation to access to it in the Java core, but it is much better to use a framework like Hibernate to access the database.

Hibernate works as an Object-Relational Mapping (ORM). This kind of frameworks provides mapping for object-oriented domain model to a relational database. Hibernate solves *object-relational impedance mismatch*(a set of conceptual and technical difficulties) problems by replacing direct persistence-related database accesses with high-level object handling functions. In other words, what Hibernate does is mapping from Java classes to database tables and vice versa.

Hibernate provides data query and retrieval facilities, and implements its own query language (*Hibernate Query Language*, HQL) in order to provide an abstraction layer for different databases, which means that working with hibernate makes the type of database transparent for the developer. Only in the configuration file must be specified the type of database that Hibernate is going to communicate with.

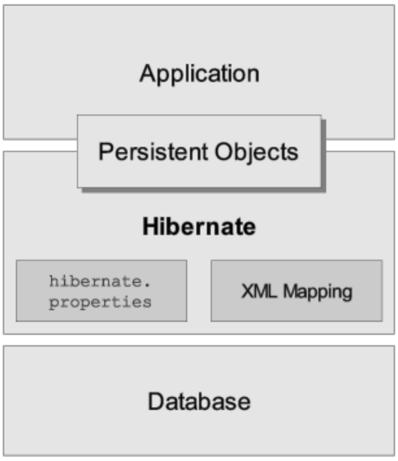


Figure 33: Schema of the Hibernate interaction

5.1.1.3 Spring security

The system implements a module of the Spring framework, which is a popular application development framework for enterprise Java. Spring framework has several modules that provide multiple functionalities, but the module used in the DAMASK project is the one that provides *authentication* and *authorization*.

The module is useful to manage the users. Actually, is not just a module but a sub-project. **Spring security** provides authentication, authorization and other security features for enterprise applications. In the DAMASK project is used to maintain the user sessions and to protect their information, mainly its password, which is encrypted using an *md5* hash.

In the spring security configuration, certain *url* patterns are defined to intercept and check for user authorization and authentication. For instance, if a user tries to access to *"/faces/profile***" spring security will intercept the request and check if the user is valid before send the request to the other controllers of the request. If the user is not valid, spring security would redirect the user to the login page or to an error page.

5.1.1.4 jQuery

jQuery is a JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid Web development. In the DAMASK project it is used for its improvement on the Javascript management and for the animations and effects, which adds a nice feel to the site.

5.1.2 Structure of the system

As explained in the above sections, the system uses a model-view-controller user interface. This strictly separates the parts of the system represented in the following schema:

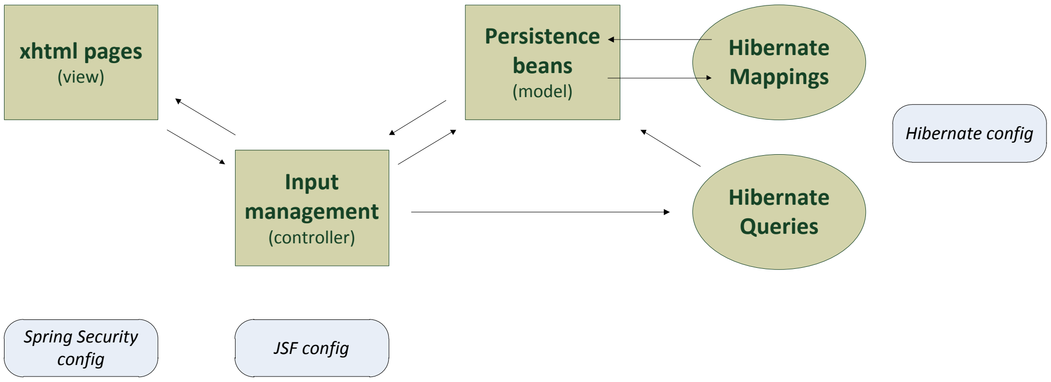


Figure 34: Schema of the system

- **xhtml pages:** JSF2 specifies that Facelets use the Extensible HyperText Markup Language (XHTML) format. Hence, the pages are a combination of xhmtl, JSF2 tags, JavaServer Pages Standard Tag Library (JSTL) and Javascript. This represents the view of the system along with the Java Beans that directly interact with the pages. These Javas beans are responsible of provide and receive information to the pages.
- **Input management:** Once an event is triggered in the view, the view manages the event and asks the controller to do something with the data gathered in the view. This usually implies two kinds of procedures (or both at once):
    - *Save data:* The user inputs data in the view that is passed to the controller to be saved in the database as a model.
    - *Retrieve data:* The view asks the controller for same data. The controller access the database to get the requested data.

  In both cases, the controller can use the data to get the desired result. For instance, the user may ask through the view about the nearest cluster to the user profile. Then the controller will get from the view the attributes percentages, checked concepts and the filter values to build the user profile. Afterwards the controller will retrieve from the database all the clusters data and check which cluster is the one that its centroid is the nearest to the user profile. Then the controller filters the selected cluster and sends the result to the view, so that the user can see the resulting cities.
- **Persistence beans:** The Hibernate framework establishes relations between rows in tables in a database and Java beans through the hibernate mappings. So, the Java beans are, in some way, the representation of the database. When the controller wants to save data into the database, it has to create persistence beans first with that data. The Hibernate queries are used to retrieve these beans (usually lists of them) with the help of HQL for filtering purposes.

5.2 A Web recommender system for touristic destinations

The DAMASK Web recommender system has been designed and implemented to be easy to use. Actually, it is prepared to be used by any kind of user, not requiring any knowledge about using other recommender systems or any specialized computer program. Hence, the Web follows the typical pattern of Web applications at Internet: a header, a menu, the body, and footer, as displayed in Figure 35.

The Web application is structured into 4 sections: a city list, a clusters list (each with its own list of cities), a profile where the user creates his personal profile and finally a page where the recommendation to the user is presented. The Web also provides a user login system in order for each user to maintain its own profile.

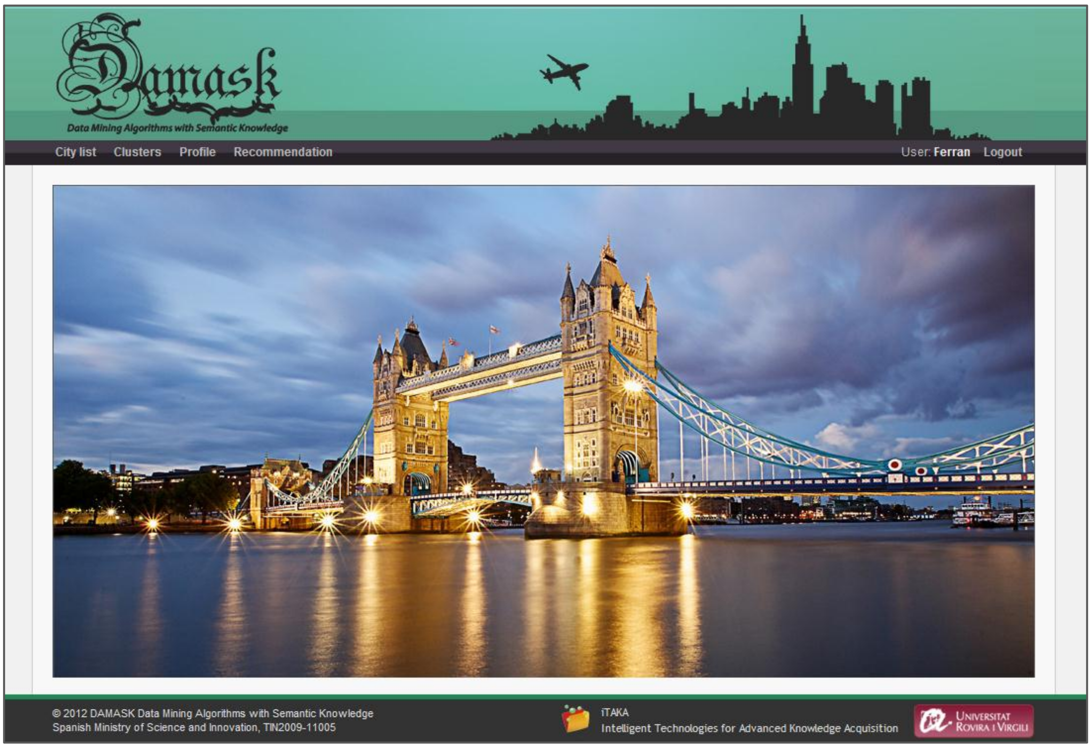

Damask
Data Mining Algorithms with Semantic Knowledge
City list
Clusters
Profile
Recommendation
User: Ferran
Logout
© 2012 DAMASK Data Mining Algorithms with Semantic Knowledge
Spanish Ministry of Science and Innovation, TIN2009-11005
iTAKA
Intelligent Technologies for Advanced Knowledge Acquisition
Universitat Rovira i Virgili

Figure 35: View of the main page

5.2.1 Parts of the Web page

The Web page presented to the user is divided into four main parts:

- The header only shows the logo of the project and an image representing the skyline of a city with a plane. The logo is used as a return-to-index link. The skyline image has been designed for decoration purposes of this prototype, devoted to the recommendation of touristic city destinations.

Figure 36: Header of the Web page in the DAMASK recommender system

- The menu part is represented as a simple bar below the header. The links to the different sections of the site (*city list*, *clusters*, *profile* and *recommendation*) are simple text links on the left, with no images that change their colors when the mouse moves over. On the right one can find the user management related links.

When not logged in, a user will see the links for either **log in** or, if it has no registered user yet, **register** a new one. When the user is logged in, the links change to a text showing the **user name** and a link to **logout**. Also, when not logged in, the available links to the sections are just the ones to access the *city list* and the *clusters list*. This is done this way because it an unregistered user cannot have a saved profile, and, without a profile, the system cannot make recommendations.

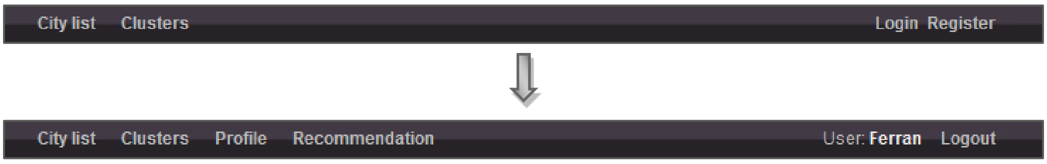


Figure 37: Menu differences when the user is not looged in and when it is.

- The content (or body) is the part of the Web that changes on every section of the menu. Hence each section will be explained in the following sections. The content for the main page is a photo of the Tower Bridge in London, just for decoration purposes. This image is licensed under Creative *Commons Attribution*.
- The footer is used to put the rights of the DAMASK project and the logos of the university and the research group where it has been developed.

© 2012 DAMASK Data Mining Algorithms with Semantic Knowledge
Spanish Ministry of Science and Innovation, TIN2009-11005

iTAKA
Intelligent Technologies for Advanced Knowledge Acquisition

Figure 38: Footer of the Web

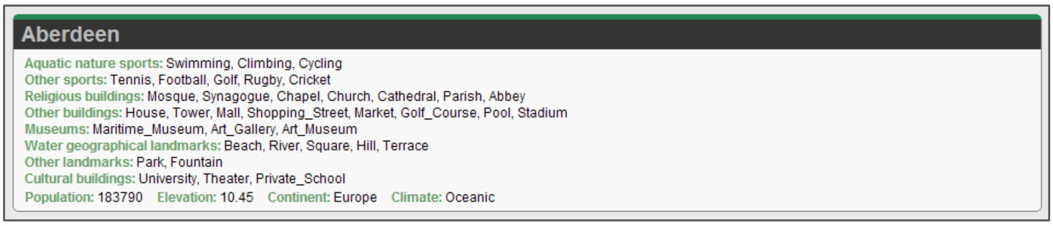
5.2.2 Sections of the Web

The application has four sections corresponding to the functionalities provided by the system. These are found in the menu bar. Each section has a menu item that opens a new Web page.

5.2.2.1 City list

In this page, a user can see the entire list of cities that have been included in the system. The cities are the 148 leading and most dynamic cities in terms of tourist arrivals, to the ranking made by Euromonitor International. For further information, see chapter 2. Actually, the cities were 150, but two were almost without information and were discarded.

Each city is represented as a table. The header of the table has the name of the city in bigger font size and different colours. Then in the content of the table, each row is each one of the semantic attributes with the list of concepts that the city has. The last row represents the numerical (population and elevation) and categorical (continent and climate) values of the city.

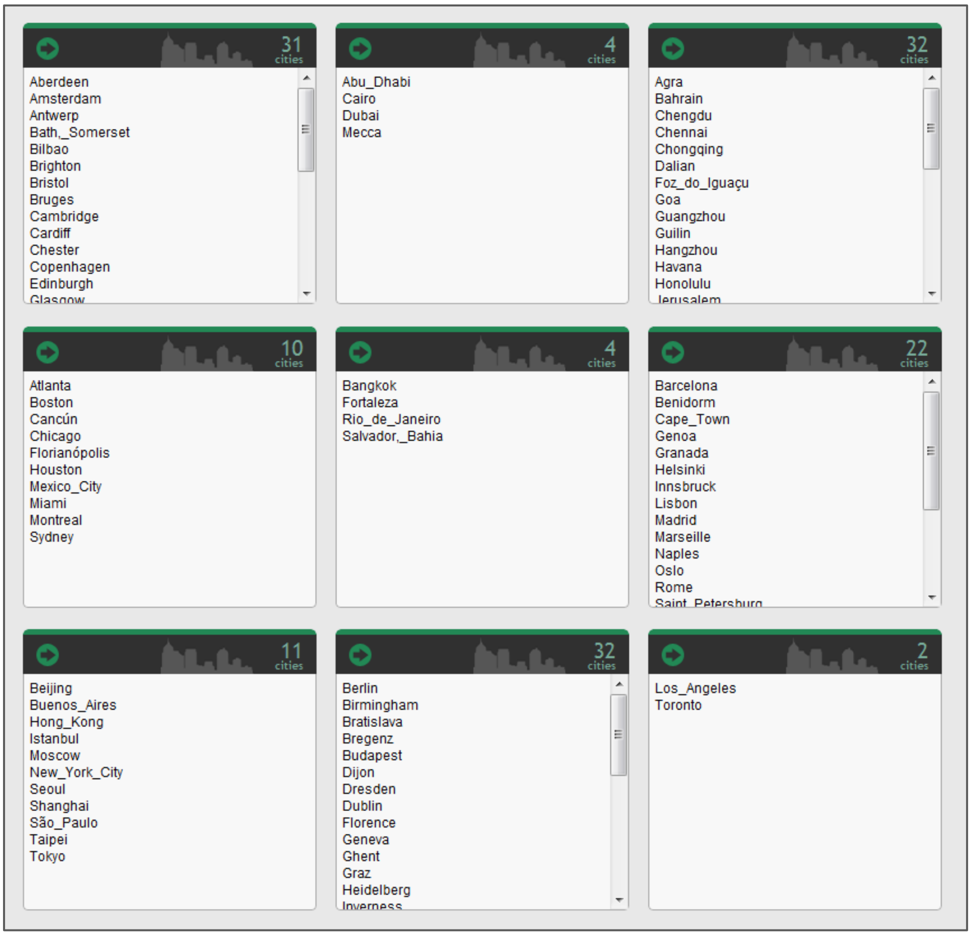


Aberdeen

Aquatic nature sports: Swimming, Climbing, Cycling
Other sports: Tennis, Football, Golf, Rugby, Cricket
Religious buildings: Mosque, Synagogue, Chapel, Church, Cathedral, Parish, Abbey
Other buildings: House, Tower, Mall, Shopping_Street, Market, Golf_Course, Pool, Stadium
Museums: Maritime_Museum, Art_Gallery, Art_Museum
Water geographical landmarks: Beach, River, Square, Hill, Terrace
Other landmarks: Park, Fountain
Cultural buildings: University, Theater, Private_School
Population: 183790 Elevation: 10.45 Continent: Europe Climate: Oceanic

Figure 39: Representation of a city in the DAMASK Web

5.2.2.2 Clusters



31 cities
Aberdeen
Amsterdam
Antwerp
Bath,_Somerset
Bilbao
Brighton
Bristol
Bruges
Cambridge
Cardiff
Chester
Copenhagen
Edinburgh
Glasgow

4 cities
Abu_Dhabi
Cairo
Dubai
Mecca

32 cities
Agra
Bahrain
Chengdu
Chennai
Chongqing
Dalian
Foz_do_Iguaçu
Goa
Guangzhou
Guilin
Hangzhou
Havana
Honolulu
Jerusalem

10 cities
Atlanta
Boston
Cancún
Chicago
Florianópolis
Houston
Mexico_City
Miami
Montreal
Sydney

4 cities
Bangkok
Fortaleza
Rio_de_Janeiro
Salvador,_Bahia

22 cities
Barcelona
Benidorm
Cape_Town
Genoa
Granada
Helsinki
Innsbruck
Lisbon
Madrid
Marseille
Naples
Oslo
Rome
Saint_Petersburg

11 cities
Beijing
Buenos_Aires
Hong_Kong
Istanbul
Moscow
New_York_City
Seoul
Shanghai
São_Paulo
Taipei
Tokyo

32 cities
Berlin
Birmingham
Bratislava
Bregenz
Budapest
Dijon
Dresden
Dublin
Florence
Geneva
Ghent
Graz
Heidelberg
Inverness

2 cities
Los_Angeles
Toronto

Figure 40: Representation of the clusters in the Web

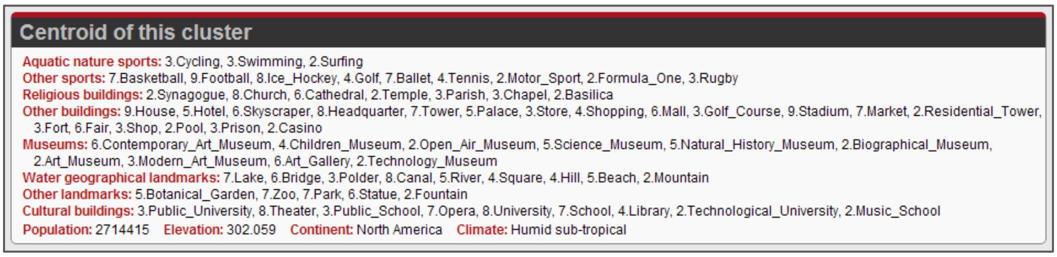
In the cluster section, the user can see 9 blocks representing the 9 clusters of the system. Each cluster has the list of their cities (just the name, not the explicit table like the one seen above). The number of cities that the cluster has is also represented (Figure 40).

In the clusters with a large list of cities, a scroll bar appears. This is done to maintain the same size in all the blocks, which results in a nicer view than having blocks of different sizes of cropped lists of cities.

One important thing here is that if the user wants to see the cities of a cluster at detail, he can click the header of a cluster (that works as a link) to access to a different page that lists the cities of the cluster in a similar way that the *city list* section does.

This cluster’s city list page is no different than the city list except for two things: (1) the main difference is that the cities are limited, obviously, to the ones that belong to the selected clusters, (2) the other difference is that at the top of the page appears the centroid of the cluster. This centroid is represented as the rest of cities, with a table with the semantic attributes in rows and the numerical and categorical attributes at the bottom, but with the difference that the semantic attributes indicate also the weight assigned to them. The weights are presented without the normalization, so that they correspond to the frequency of appearance of the concepts in the cluster (see Figure 41).

In order to facilitate the identification of the centroid, the colour is different than the one used for the list of cities (see Figure 42).

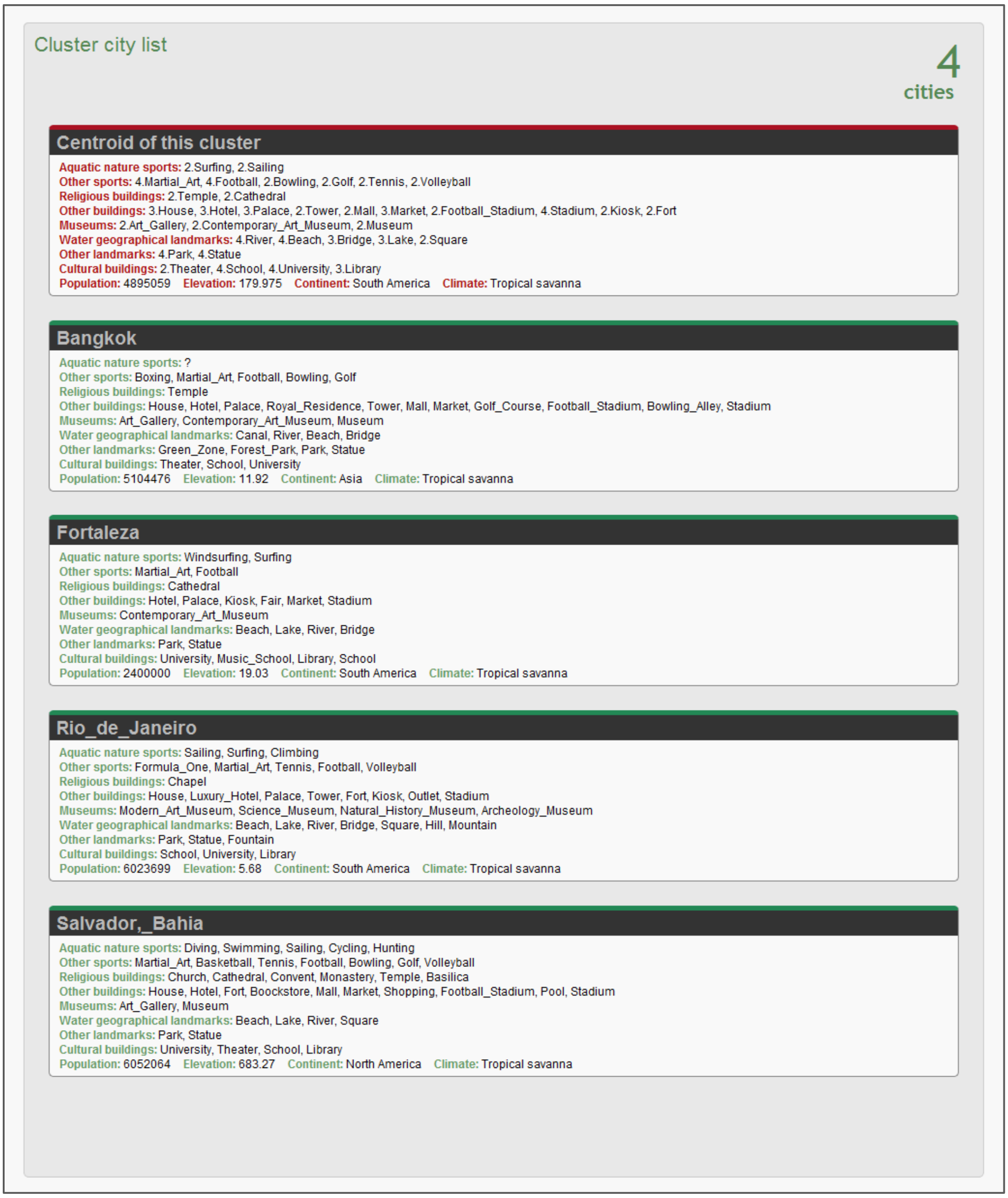


Centroid of this cluster

Aquatic nature sports: 3.Cycling, 3.Swimming, 2.Surfing
Other sports: 7.Basketball, 9.Football, 8.Ice_Hockey, 4.Golf, 7.Ballet, 4.Tennis, 2.Motor_Sport, 2.Formula_One, 3.Rugby
Religious buildings: 2.Synagogue, 8.Church, 6.Cathedral, 2.Temple, 3.Parish, 3.Chapel, 2.Basilica
Other buildings: 9.House, 5.Hotel, 6.Skyscraper, 8.Headquarter, 7.Tower, 5.Palace, 3.Store, 4.Shopping, 6.Mall, 3.Golf_Course, 9.Stadium, 7.Market, 2.Residential_Tower, 3.Fort, 6.Fair, 3.Shop, 2.Pool, 3.Prison, 2.Casino
Museums: 6.Contemporary_Art_Museum, 4.Children_Museum, 2.Open_Air_Museum, 5.Science_Museum, 5.Natural_History_Museum, 2.Biographical_Museum, 2.Art_Museum, 3.Modern_Art_Museum, 6.Art_Gallery, 2.Technology_Museum
Water geographical landmarks: 7.Lake, 6.Bridge, 3.Polder, 8.Canal, 5.River, 4.Square, 4.Hill, 5.Beach, 2.Mountain
Other landmarks: 5.Botanical_Garden, 7.Zoo, 7.Park, 6.Statue, 2.Fountain
Cultural buildings: 3.Public_University, 8.Theater, 3.Public_School, 7.Opera, 8.University, 7.School, 4.Library, 2.Technological_University, 2.Music_School
Population: 2714415 Elevation: 302.059 Continent: North America Climate: Humid sub-tropical

Figure 41: Representation of the centroid of a cluster.

The next image represents the view of an entire cluster. The idea is that the user can compare the concepts on each of the cities and the centroid that represents the cluster. Notice that at the top of the page, a number appears telling the user the number of cities that the cluster has.



Cluster city list

4
cities

Centroid of this cluster

Aquatic nature sports: 2.Surfing, 2.Sailing

Other sports: 4.Martial_Art, 4.Football, 2.Bowling, 2.Golf, 2.Tennis, 2.Volleyball

Religious buildings: 2.Temple, 2.Cathedral

Other buildings: 3.House, 3.Hotel, 3.Palace, 2.Tower, 2.Mall, 3.Market, 2.Football_Stadium, 4.Stadium, 2.Kiosk, 2.Fort

Museums: 2.Art_Gallery, 2.Contemporary_Art_Museum, 2.Museum

Water geographical landmarks: 4.River, 4.Beach, 3.Bridge, 3.Lake, 2.Square

Other landmarks: 4.Park, 4.Statue

Cultural buildings: 2.Theater, 4.School, 4.University, 3.Library

Population: 4895059 Elevation: 179.975 Continent: South America Climate: Tropical savanna

Bangkok

Aquatic nature sports: ?

Other sports: Boxing, Martial_Art, Football, Bowling, Golf

Religious buildings: Temple

Other buildings: House, Hotel, Palace, Royal_Residence, Tower, Mall, Market, Golf_Course, Football_Stadium, Bowling_Alley, Stadium

Museums: Art_Gallery, Contemporary_Art_Museum, Museum

Water geographical landmarks: Canal, River, Beach, Bridge

Other landmarks: Green_Zone, Forest_Park, Park, Statue

Cultural buildings: Theater, School, University

Population: 5104476 Elevation: 11.92 Continent: Asia Climate: Tropical savanna

Fortaleza

Aquatic nature sports: Windsurfing, Surfing

Other sports: Martial_Art, Football

Religious buildings: Cathedral

Other buildings: Hotel, Palace, Kiosk, Fair, Market, Stadium

Museums: Contemporary_Art_Museum

Water geographical landmarks: Beach, Lake, River, Bridge

Other landmarks: Park, Statue

Cultural buildings: University, Music_School, Library, School

Population: 2400000 Elevation: 19.03 Continent: South America Climate: Tropical savanna

Rio_de_Janeiro

Aquatic nature sports: Sailing, Surfing, Climbing

Other sports: Formula_One, Martial_Art, Tennis, Football, Volleyball

Religious buildings: Chapel

Other buildings: House, Luxury_Hotel, Palace, Tower, Fort, Kiosk, Outlet, Stadium

Museums: Modern_Art_Museum, Science_Museum, Natural_History_Museum, Archeology_Museum

Water geographical landmarks: Beach, Lake, River, Bridge, Square, Hill, Mountain

Other landmarks: Park, Statue, Fountain

Cultural buildings: School, University, Library

Population: 6023699 Elevation: 5.68 Continent: South America Climate: Tropical savanna

Salvador,_Bahia

Aquatic nature sports: Diving, Swimming, Sailing, Cycling, Hunting

Other sports: Martial_Art, Basketball, Tennis, Football, Bowling, Golf, Volleyball

Religious buildings: Church, Cathedral, Convent, Monastery, Temple, Basilica

Other buildings: House, Hotel, Fort, Boockstore, Mall, Market, Shopping, Football_Stadium, Pool, Stadium

Museums: Art_Gallery, Museum

Water geographical landmarks: Beach, Lake, River, Square

Other landmarks: Park, Statue

Cultural buildings: University, Theater, School, Library

Population: 6052064 Elevation: 683.27 Continent: North America Climate: Tropical savanna

Figure 42: Detail view of a cluster with the centroid and its list of cities.

5.2.2.3 Profile

This section is only available for registered users that are logged in. This page provides some options for a user to build a personal profile, which will be used to make recommendations to the user based on the similarity with the centroid of the cluster.

In this section we first present the details about the structure and contents of the user profile that have been defined in the project. Secondly, the interface through the Web page is detailed.

5.2.2.3.1 Structure of the user profile

The different alternatives available in the system, the objects, are defined using three types of data: numerical, categorical and semantic attributes. Therefore, the user must be able to express some kind of preference information regarding each of those types of attributes. The requirements can be given in two forms:

a) Mandatory constraints or filters: the user indicates the subset of values he is interested on. Any value outside this subset is discarded.

b) Preference requirements: the user indicates which values are the ones he prefers, but without discarding similar values.

Each application domain will require a different type of requirement for each attribute. In the case study in the DAMASK project about selection of touristic destinations, the semantic attributes are the criteria that define the type of destination desired by the user, while the numerical and categorical attributes refer to contextual information, so they can be better used as filtering criteria. The following table summarizes the criteria and its role in the user profile. For each of the attributes, the range of possible values is also indicated.

Table 18: Attributes by types and its values for the user profile.

Preference attributes	Type	Values
Aquatic nature sports	S	Swimming, Rafting, Surfing, Diving, Kayaking, Skiing, Snowboarding, Climbing, Mountain_Biking, Cycling, Water_Polo, Windsurfing, Waterskiing, Hunting, Sailing
Other sports	S	Football, Squash, Roller_Hockey, Formula_One, Skateboarding, Hockey, Boxing, Silat, Basketball, Rally, Popular_Running, Judo, Golf, Bowling, Swimming_Race, Stickball, Motor_Sport, Badminton, Volleyball, Martial_Art, Ballet, Handball, Street_Hockey, Rugby, Tennis, Karate, Table_Tennis, Cricket, Ice_Hockey, Paddle
Religious buildings	S	Synagogue, Mosque, Chapel, Sanctuary, Temple, Cathedral, Basilica, Religious_Building, Abbey, Church, Parish, Convent, Monastery
Other buildings	S	Casino_Resort, Stadium, Flea_Market, Mall, Cotton_Mill, Residential_District, Trade_Fair, Royal_Residence, Formula_One_Circuit, Golf_Course, Skyscraper, Football_Stadium, Velodrome, Headquarter, House, Prison, Shopping_Centre, Kiosk, Pool, Souvenir_Shop, Shop, Sport_Complex, Residential_Building, Hockey_Stadium, Townhouse, Market, Palace, Residential_Tower, Store, Outlet, Luxury_Hotel, Shophouse, Fair, Tower, Boockstore, Industrial_Building, Food_Market, Fashion_Shop, Aquatic_Centre, Fort, Shopping_Street, Supermarket, Shopping, Squash_Centre, Camping, Baseball_Stadium, Casino, Hotel, Hypermarket, Convenience_Store, Garden_Apartments, Bowling_Alley, Antiquarian_Shop, Shopping_Area
Museums	S	Modern_Art_Museum, Erotic_Museum, Egyptian_Museum, Toy_Museum, Open_Air_Museum, Railway_Museum, Fishing_Museum, Music_Museum, Wax_Museum, Biographical_Museum, Folk_Art_Museum, Aviation_Museum, Maritime_Museum, Military_Museum, Technology_Museum, Sex_Museum, Art_Gallery, Archeology_Museum, Art_Museum, Science_Museum, Contemporary_Art_Museum, Museum, Industrial_Museum, Astronomy_Museum, Children_Museum, Natural_History_Museum, Woman_Museum, Computer_Museum
Water geographical landmarks	S	Hill, Stone_Bridge, Square, Polder, Bridge, Cave, Terrace, Pedestrian_Bridge, Canal, Gorge, Beach, Lake, River, Mountain
Other landmarks	S	Tomb, Sepulchre, Obelisk, Park, Urban_Park, Statue, Garden_Park, Nature_Reserve, Ionic_Column, Zoo, Pyramid, Ancient_Obelisk, Historic_Park, Column, Fountain, Megalithic, Green_Zone, Refuge, Forest_Park, Crypt, Egyptian_Obelisk, Mausoleum, Suburban_Park, Botanical_Garden
Cultural buildings	S	Private_School, Opera, Ancient_Greek_Theatre, Business_School, Forum, Public_School, Music_School, University, Private_University, Public_University, Library, Roman_Amphitheatre, Technological_University, Theater, School,

		Art_School, Amphitheatre
Filter attributes		
Population	N	0 - 15.000.000
Elevation	N	0 - 2.500
Continent	C	Europe, Asia, Africa, North America, South America, Oceania
Climate	C	Alpine, Desert, Humid continental, Humid sub-tropical, Mediterranean, Oceanic, Polar, Semi-arid, Subarctic, Tropical monsoon, Tropical rainforest, Tropical savanna

For each of the filtering criteria, the user will determine the range of accepted values. For example to indicate low values for the Population if one wants to be in a small city.

For the preference criteria, two types of information must be given:

- The *weight*, interpreted as the relative preference of that attribute with respect to the other.
- The *preferred values*, a list of terms (semantic concepts) that fit with the user's interests.

5.2.2.3.2 Definition of the profile in the Web page

The Web page has two differentiated sections: one with the semantic attributes to indicate the user's preference profile, and one with the numerical and categorical attributes that work as a filter for the recommendation result.

The part with the preference semantic criteria is presented with sliders, one for each one of the criteria, which are used to indicate the degree of importance that the user wants to give to the attribute (Figure 43). In other words, the user will set a preference between 0% and 100% for each semantic attribute. This weight is used when the similarity with the clusters' centroid is computed.

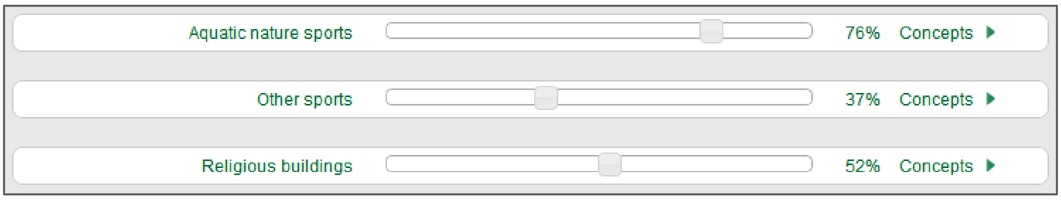
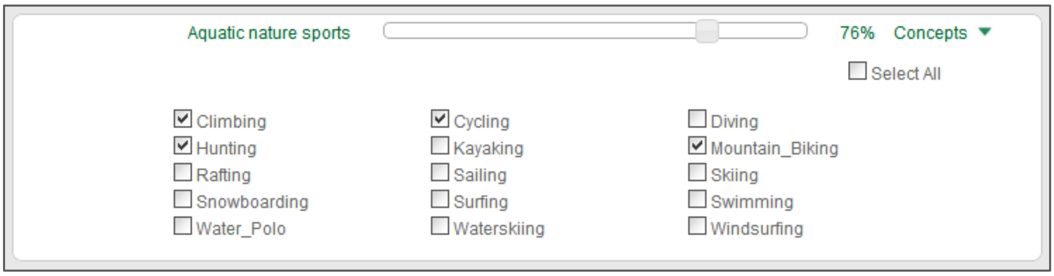


Figure 43: Respresentation of the semantic attributes preference

In addition, the user has to introduce which are the values that we is looking for in the city. The user can click the "concepts ►" link in order to unfold the corresponding section and see the list of concepts available to check. The system builds a list with the selected concepts, which is compared against the list in clusters' centroids. This process is explained in the next section. An extra checkbox is provided as a *select all* / *unselect all* toggle mechanism, as shown in Figure 44.



Aquatic nature sports 76% Concepts

□ Select All

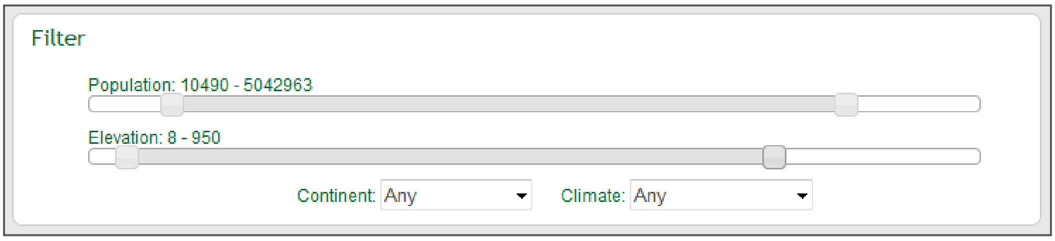
☑ Climbing ☑ Cycling □ Diving
☑ Hunting □ Kayaking ☑ Mountain_Biking
□ Rafting □ Sailing □ Skiing
□ Snowboarding □ Surfing □ Swimming
□ Water_Polo □ Waterskiing □ Windsurfing

Figure 44: Representation of a semantic attribut list of concepts

If some criterion is set to 0%, it indicates that the user will not give any preference information regarding this attribute, so it is not used in the comparison with the centroids of the clusters.

The other part of the same page is the filtering section, used to reduce the resulting list of recommended cities, just keeping the cities that fulfill the constraints indicated by these criteria.

The numerical filters are implemented with two range sliders, whereas two selection lists elements are used for categorical attributes (Figure 45).



Filter

Population: 10490 - 5042963

Elevation: 8 - 950

Continent: Any Climate: Any

Figure 45: Representation of the filter in the profile page.

There is one interesting thing here, the values of the sliders do not increase in an arithmetic progression way, but they do it following a geometric progression instead. Notice that the city with the lowest population has only 20.000 people, and the biggest has about 15.000.000. The difference is enormous, and because of that, the change of the value of the slider for just 1 pixel is also big. This caused, in a first implementation, that the range of the majority of cities was too small. The values rapidly passed from 0 to 50.000 in just one pixel or two, what is horrible when 60 cities, the 40% of the total cities, are in that range. The same occurs with the Elevation since the majority of the most touristic cities are placed in the coast. Remember the graphics of frequency distribution surveyed at chapter 2:

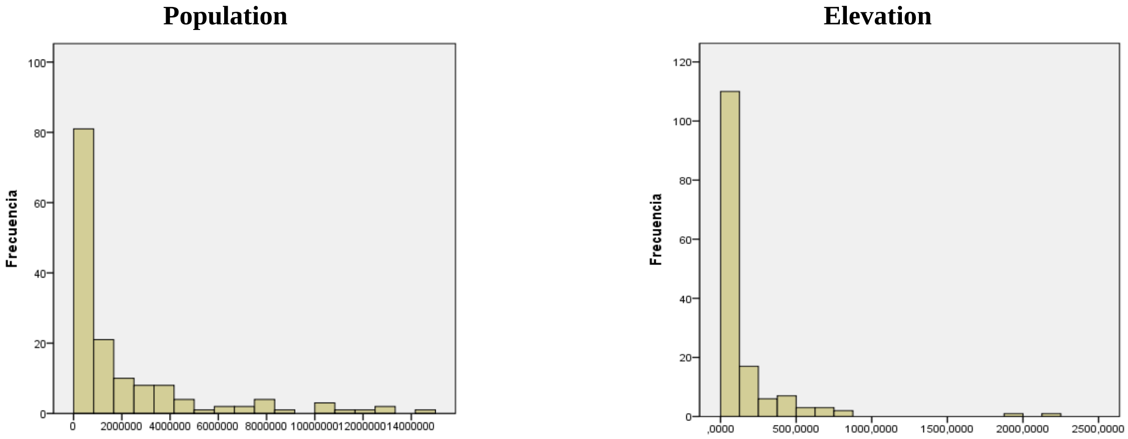


Figure 46: Frequency distribution of the numerical variables

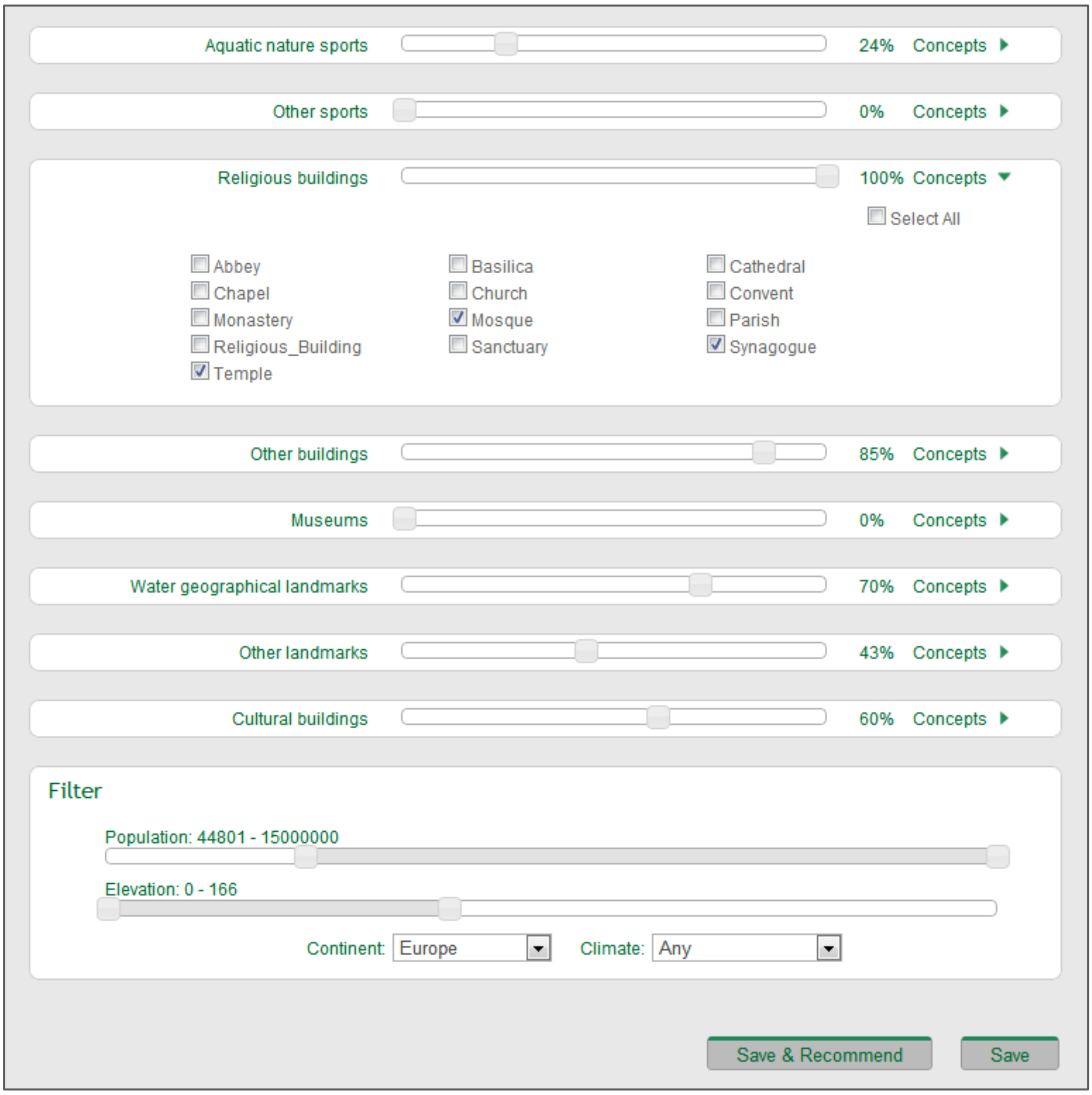
Hence, to solve this issue, we have implemented a function to transform the arithmetic progression of the sliders provided by jQuery into a more suitable geometric progression. With this, the values of the slider are better distributed. Starting with a change of just 1000 and ending with changes of about 500.000.

The select elements for the continent and the climate show all possible values plus the "*Any*" option, that is selected as default. The selection of "*Any*" indicates that the used does not want to use the *filter* option, so that all the values are accepted.

Finally, there are two buttons at the bottom of the page, "*Save*" and "*Save and recommend*". The first one is just to save the profile of the user in the database. The second does the same, but redirects to the recommendation page automatically (which is the expected flow of events).

Figure 47: Buttons of the profile page.

The following image is an entire view of the profile page. An attribute was unfolded to see its concepts. Some of these concepts were checked. The filters were also changed in a manner that the recommendation will filter for European cities with population above 45.000 and elevation below 166. The list of cities in the recommendation will have any kind of climate since "*Any*" is the selected option.



Aquatic nature sports 24% Concepts ▸

Other sports 0% Concepts ▸

Religious buildings 100% Concepts ▾

□ Select All

□ Abbey
□ Chapel
□ Monastery
□ Religious_Building
☑ Temple

□ Basilica
□ Church
☑ Mosque
□ Sanctuary

□ Cathedral
□ Convent
□ Parish
☑ Synagogue

Other buildings 85% Concepts ▸

Museums 0% Concepts ▸

Water geographical landmarks 70% Concepts ▸

Other landmarks 43% Concepts ▸

Cultural buildings 60% Concepts ▸

Filter

Population: 44801 - 15000000

Elevation: 0 - 166

Continent: Europe Climate: Any

Figure 48: View of the profile page with an unfold attribute and some concepts checked.

5.2.2.4 Recommendation

Once the user has completed its profile both with the preference information and the filtering criteria, the system starts the recommendation process. If a user tries to access the recommendation page without completing his/her personal profile, the system will redirect the user automatically to the profile page.

5.2.2.4.1 The recommendation algorithm

```
Retrieve from database the user profile
Retrieve from database all the clusters centroids
For each centroid {
    For each attribute of the centroid {
        Compute the distance from the centroid value to the user profile value
            multiplying the distance by the weight of the attribute in the user
            profile
    }
    Sum all the weighted distances to obtain the absolute distance between the
        user profile and the centroid
}
Select the centroid with the lowest distance and retrieve its cluster from DB
Filter the cities of this cluster according to the ranges given in the user’s profile
The clusters of the centroids whose distance is below a threshold are also
    filtered and saved as “additional cities” set
Show the results to the user
```

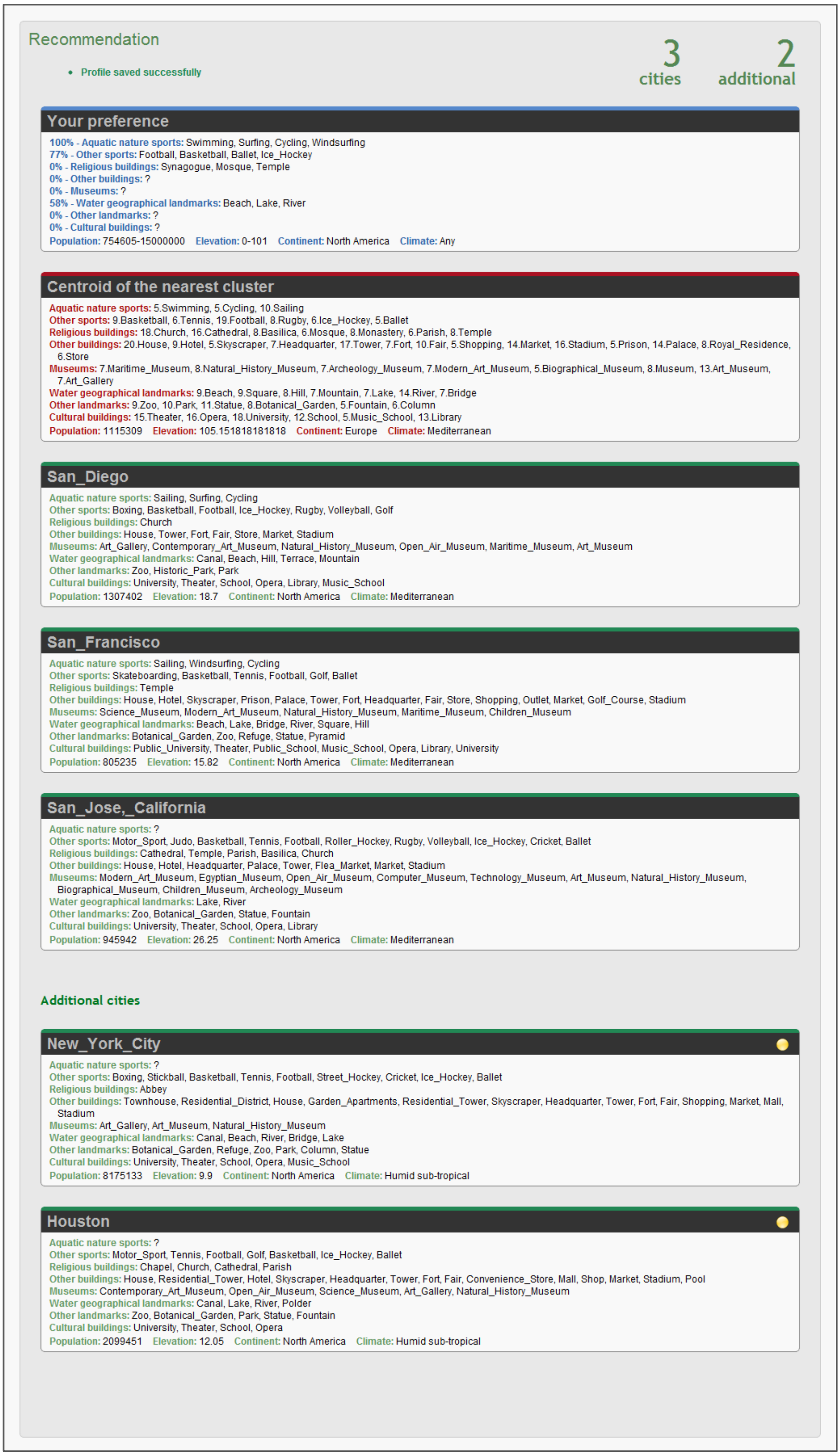
The recommendation process starts creating the prototype of a city from the concepts selected in the profile page. This prototype only has semantic attributes since the numerical and categorical are reserved to filtering purposes.

Then, the distance between the prototype and the centroids of the different clusters is computed in a similar way that it has been done in the clustering application. The prototype works as a plain city in this procedure. When the algorithm is at the point where the distance is computed for each semantic attribute, each of these distances is multiplied by the factor that the user specified in his profile. After that, the semantic distance is computed. This distance works here as the definitive distance, because the numerical and categorical distances are not computed.

The following step is devoted to check which centroid is the most similar to the prototype. The selected centroid is used then to get the cluster that represents. Finally, the list of cities of the cluster is filtered using the user’s profile indications.

5.2.2.4.2 Recommendation of touristic city destinations in the DAMASK Web application

The results are displayed in a similar way than the list of cities with their description (option City List). However, we find some particularities.



Recommendation

- Profile saved successfully

3 cities   2 additional

Your preference

100% - Aquatic nature sports: Swimming, Surfing, Cycling, Windsurfing
77% - Other sports: Football, Basketball, Ballet, Ice_Hockey
0% - Religious buildings: Synagogue, Mosque, Temple
0% - Other buildings: ?
0% - Museums: ?
58% - Water geographical landmarks: Beach, Lake, River
0% - Other landmarks: ?
0% - Cultural buildings: ?
Population: 754605-15000000   Elevation: 0-101   Continent: North America   Climate: Any

Centroid of the nearest cluster

Aquatic nature sports: 5.Swimming, 5.Cycling, 10.Sailing
Other sports: 9.Basketball, 6.Tennis, 19.Football, 8.Rugby, 6.Ice_Hockey, 5.Ballet
Religious buildings: 18.Church, 16.Cathedral, 8.Basilica, 6.Mosque, 8.Monastery, 6.Parish, 8.Temple
Other buildings: 20.House, 9.Hotel, 5.Skyscraper, 7.Headquarter, 17.Tower, 7.Fort, 10.Fair, 5.Shopping, 14.Market, 16.Stadium, 5.Prison, 14.Palace, 8.Royal_Residence, 6.Store
Museums: 7.Maritime_Museum, 8.Natural_History_Museum, 7.Archeology_Museum, 7.Modern_Art_Museum, 5.Biographical_Museum, 8.Museum, 13.Art_Museum, 7.Art_Gallery
Water geographical landmarks: 9.Beach, 9.Square, 8.Hill, 7.Mountain, 7.Lake, 14.River, 7.Bridge
Other landmarks: 9.Zoo, 10.Park, 11.Statue, 8.Botanical_Garden, 5.Fountain, 6.Column
Cultural buildings: 15.Theater, 16.Opera, 18.University, 12.School, 5.Music_School, 13.Library
Population: 1115309   Elevation: 105.151818181818   Continent: Europe   Climate: Mediterranean

San_Diego

Aquatic nature sports: Sailing, Surfing, Cycling
Other sports: Boxing, Basketball, Football, Ice_Hockey, Rugby, Volleyball, Golf
Religious buildings: Church
Other buildings: House, Tower, Fort, Fair, Store, Market, Stadium
Museums: Art_Gallery, Contemporary_Art_Museum, Natural_History_Museum, Open_Air_Museum, Maritime_Museum, Art_Museum
Water geographical landmarks: Canal, Beach, Hill, Terrace, Mountain
Other landmarks: Zoo, Historic_Park, Park
Cultural buildings: University, Theater, School, Opera, Library, Music_School
Population: 1307402   Elevation: 18.7   Continent: North America   Climate: Mediterranean

San_Francisco

Aquatic nature sports: Sailing, Windsurfing, Cycling
Other sports: Skateboarding, Basketball, Tennis, Football, Golf, Ballet
Religious buildings: Temple
Other buildings: House, Hotel, Skyscraper, Prison, Palace, Tower, Fort, Headquarter, Fair, Store, Shopping, Outlet, Market, Golf_Course, Stadium
Museums: Science_Museum, Modern_Art_Museum, Natural_History_Museum, Maritime_Museum, Children_Museum
Water geographical landmarks: Beach, Lake, Bridge, River, Square, Hill
Other landmarks: Botanical_Garden, Zoo, Refuge, Statue, Pyramid
Cultural buildings: Public_University, Theater, Public_School, Music_School, Opera, Library, University
Population: 805235   Elevation: 15.82   Continent: North America   Climate: Mediterranean

San_Jose,_California

Aquatic nature sports: ?
Other sports: Motor_Sport, Judo, Basketball, Tennis, Football, Roller_Hockey, Rugby, Volleyball, Ice_Hockey, Cricket, Ballet
Religious buildings: Cathedral, Temple, Parish, Basilica, Church
Other buildings: House, Hotel, Headquarter, Palace, Tower, Flea_Market, Market, Stadium
Museums: Modern_Art_Museum, Egyptian_Museum, Open_Air_Museum, Computer_Museum, Technology_Museum, Art_Museum, Natural_History_Museum, Biographical_Museum, Children_Museum, Archeology_Museum
Water geographical landmarks: Lake, River
Other landmarks: Zoo, Botanical_Garden, Statue, Fountain
Cultural buildings: University, Theater, School, Opera, Library
Population: 945942   Elevation: 26.25   Continent: North America   Climate: Mediterranean

Additional cities

New_York_City ●

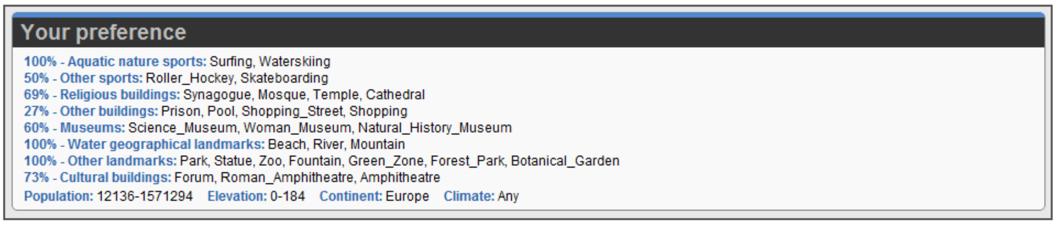
Aquatic nature sports: ?
Other sports: Boxing, Stickball, Basketball, Tennis, Football, Street_Hockey, Cricket, Ice_Hockey, Ballet
Religious buildings: Abbey
Other buildings: Townhouse, Residential_District, House, Garden_Apartments, Residential_Tower, Skyscraper, Headquarter, Tower, Fort, Fair, Shopping, Market, Mall, Stadium
Museums: Art_Gallery, Art_Museum, Natural_History_Museum
Water geographical landmarks: Canal, Beach, River, Bridge, Lake
Other landmarks: Botanical_Garden, Refuge, Zoo, Park, Column, Statue
Cultural buildings: University, Theater, School, Opera, Music_School
Population: 8175133   Elevation: 9.9   Continent: North America   Climate: Humid sub-tropical

Houston ●

Aquatic nature sports: ?
Other sports: Motor_Sport, Tennis, Football, Golf, Basketball, Ice_Hockey, Ballet
Religious buildings: Chapel, Church, Cathedral, Parish
Other buildings: House, Residential_Tower, Hotel, Skyscraper, Headquarter, Tower, Fort, Fair, Convenience_Store, Mall, Shop, Market, Stadium, Pool
Museums: Contemporary_Art_Museum, Open_Air_Museum, Science_Museum, Art_Gallery, Natural_History_Museum
Water geographical landmarks: Canal, Lake, River, Polder
Other landmarks: Zoo, Botanical_Garden, Park, Statue, Fountain
Cultural buildings: University, Theater, School, Opera
Population: 2099451   Elevation: 12.05   Continent: North America   Climate: Humid sub-tropical

Figure 49: View of the recommendation page, with some cities of the recommended cluster and some others additional.

The first *block* shown represents the user profile introduced in the system (Figure 50). For each preference semantic attribute, the relative weight is indicated in percentage, together with the selected concepts. Filters are also displayed at the bottom. Different colors are used to differentiate it from the list of recommended cities.



Your preference

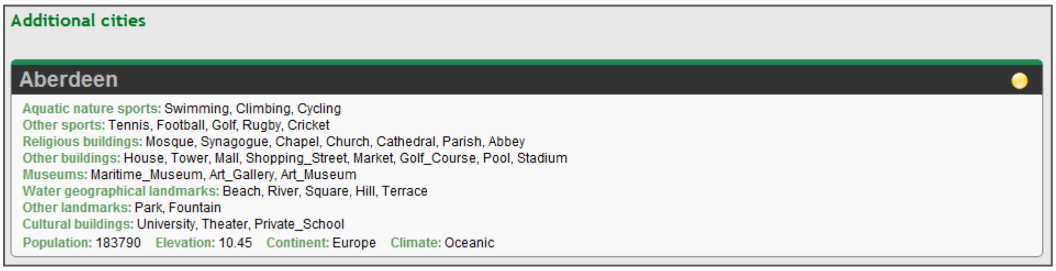
100% - Aquatic nature sports: Surfing, Waterskiing
50% - Other sports: Roller_Hockey, Skateboarding
69% - Religious buildings: Synagogue, Mosque, Temple, Cathedral
27% - Other buildings: Prison, Pool, Shopping_Street, Shopping
60% - Museums: Science_Museum, Woman_Museum, Natural_History_Museum
100% - Water geographical landmarks: Beach, River, Mountain
100% - Other landmarks: Park, Statue, Zoo, Fountain, Green_Zone, Forest_Park, Botanical_Garden
73% - Cultural buildings: Forum, Roman_Amphitheatre, Amphitheatre
Population: 12136-1571294 Elevation: 0-184 Continent: Europe Climate: Any

Figure 50: Representation of the prototype

After the user profile representation we can find the centroid of the most similar cluster (displayed as in the *clusters* option). Below we can find the list of recommended touristic cities. Only the ones that have passed the filters are displayed.

When the filters are too strict (allowing a small interval of possibilities) the list of destinations recommended may be too short. To allow the user to find some other alternatives, it has been implemented a mechanism to show additional cities, that are also similar to the user's profile. All the clusters within a distance to the profile smaller than a given threshold are considered. All the cities that belong to those clusters are filtered according to the mandatory criteria and a secondary list of alternatives is obtained.

At first, the additional cities are hidden in the Web page, and they will only be showed if the user clicks the "*Additional cities*" link at the bottom of the page. To differentiate the additional cities from the main ones, a yellow ball icon is added at the top right of the *city box*.



Additional cities

Aberdeen ●

Aquatic nature sports: Swimming, Climbing, Cycling
Other sports: Tennis, Football, Golf, Rugby, Cricket
Religious buildings: Mosque, Synagogue, Chapel, Church, Cathedral, Parish, Abbey
Other buildings: House, Tower, Mall, Shopping_Street, Market, Golf_Course, Pool, Stadium
Museums: Maritime_Museum, Art_Gallery, Art_Museum
Water geographical landmarks: Beach, River, Square, Hill, Terrace
Other landmarks: Park, Fountain
Cultural buildings: University, Theater, Private_School
Population: 183790 Elevation: 10.45 Continent: Europe Climate: Oceanic

Figure 51: Representation of an additional city and the "*Additional cities*" link.

6 Evaluation of the results

In this chapter, we make an analysis of the results of the prototype implemented in the DAMASK project, focused on recommending the most adequate touristic city destinations for a given user.

Given the database collected using the tools developed in previous works of the DAMASK project, we analyse here the results of the methods developed in this work. They are focused on the clustering method as well as the recommendation algorithm that is based on the prototypes generated by the clustering stage.

The analysis of the results is quite difficult due large dimension of the data matrix, composed by 150 cities, which cities are described by 8 semantic attributes (aquatic nature sports, other sports, religious buildings, cultural buildings, other buildings, museums, water geographical landmarks and other landmarks), 2 numerical attributes (population and elevation) and 2 categorical attributes (continent and climate). This document aims at showing the utility of the prototype implemented for the recommendation of touristic cities, using data that is available in the Web.

Although a comprehensive analysis of the final results is done at the end, first we start considering some examples that are explained just over small portions of the results, since the total resulting matrix is so huge that makes impossible to be even clearly printed.

Another difficulty of the analysis is the nature of the data, having multi-valued attributes. The data matrix is highly heterogeneous, which makes very hard to see the right patterns in the clustering results. For instance, two cities can be very similar in a pair of attributes, even exactly equal, but if the other attributes are very dissimilar, it is very likely that the two cities end up in different clusters. Since each semantic attribute may have a long list of values, the possibilities of combining concepts are enormous and this is what makes the things hard for a manual analysis, making sometimes difficult to understand the criteria used to put a city in a determined cluster. Despite of having these difficulties, a qualitative analysis of the clusters has been done showing that the semantic component of the objects has a great influence on the result, obtaining compact and interpretable clusters.

After analysing the clustering results, the document studies the recommendation process by means of different tests using the DAMASK web system.

In addition to the final configuration of the system (with the values established in the previous theoretical analysis of each of the methods), other configurations have been tested. For example, we have considered the case when the numerical and categorical weights are reduced to 0, in order to see which are the results using only semantic information.

6.1 Study of the Clustering Results

The clustering methodology is described in chapter 4. In this section, the results obtained using different values for the parameters are analysed. Parameters like the threshold or the weights applied for the different types of attributes are studied. Variations on the OWA operator used into the distance calculation (as seen in chapter 3) are also studied.

The parameters of the clustering process are the following:

- OWA: Determine the type of OWA applied at computing the distances between cities.
- K: the number of clusters.
- λ: the cut threshold.
- W_S: Weight applied to the semantic attributes.
- W_N: Weight applied to the numerical attributes.
- W_C: Weight applied to the categorical attributes.

To study the clusters, the results of the clustering have been copied from standard output of the system to an Excel file as explained in chapter 4.

The first analysis in this section is for the definitive values for the parameters that have been fixed in the Web recommender system. Afterwards, other values of those parameters are considered in order to see which changes may cause the modification of those parameters.

6.1.1 Analysis of the results in the final configuration of the system

As explained in the previous documents, the values of the parameters that are finally used in the recommender system use the following ones:

OWA: Linguistic quantifiers (*most*), K = 10, $\lambda = 0.2$, N = 150, W_S= 0.7, $W_N = 0.15$, $W_C = 0.15$.

It is important to explain that the resulting excel has some columns that are used to see how much distance there is between each city and the centroid for their absolute distance, semantic distance, numerical distance and categorical distance. In table 19 can be seen an example of these columns. It is worth to be remarked that the *absolute weight* depends on the weights applied to each one of the attributes. In the final configuration of the system, these weights are 0.7 for the semantic distance, 0.15 for the numerical distance and 0.15 for the categorical distance.

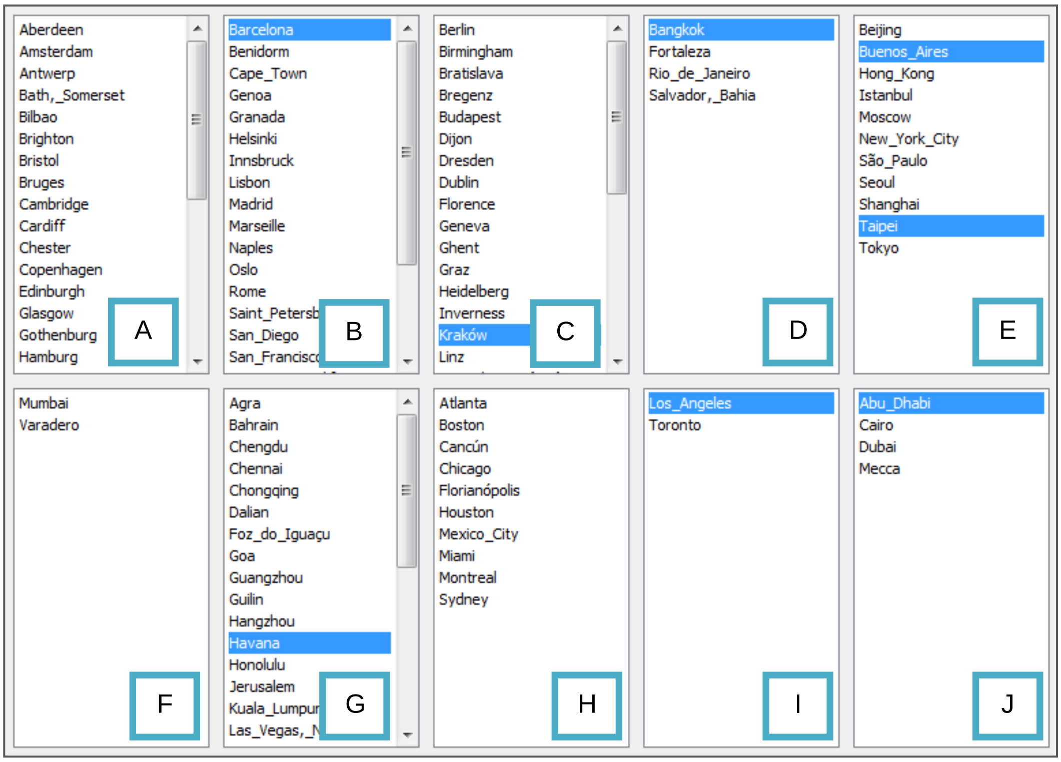
Table 19: Representation of the columns that indicate the absolute, semantic, numerical and categorical distances for each city to their cluster's centroid.

	Absolute D.	Semantic D.	Numerical D.	Categorical D.
Centroid				
Bangkok	0,26	0,18	0,36	0,50
Fortaleza	0,19	0,15	0,59	0,00
Rio_de_Janeiro	0,17	0,15	0,43	0,00
Salvador,_Bahia	0,24	0,12	0,59	0,50
Centroid				
Abu_Dhabi	0,20	0,19	0,44	0,00
Cairo	0,34	0,20	0,85	0,50
Dubai	0,20	0,22	0,35	0,00
Mecca	0,29	0,29	0,58	0,00
Centroid				
Atlanta	0,17	0,15	0,44	0,00
Boston	0,20	0,14	0,68	0,00
Cancún	0,53	0,50	0,69	0,50
Chicago	0,15	0,06	0,26	0,50

Florianópolis	0,40	0,31	0,70	0,50
Houston	0,18	0,13	0,56	0,00
Mexico_City	0,30	0,11	1,00	0,50
Miami	0,28	0,15	0,70	0,50
Montreal	0,20	0,13	0,22	0,50
Sydney	0,25	0,11	0,64	0,50

With these values on the parameters, the clustering algorithm generates a partition in 10 clusters with different sizes, but with a maximum of around 30 cities in the same cluster (the average would be 150/10 = 15), twice the expected average size.

Figure 52 shows the clusters, whose corresponding sizes are: 31, 22, 32, 4, 11, 2, 32, 10, 2 and 4.



Aberdeen
Amsterdam
Antwerp
Bath,_Somerset
Bilbao
Brighton
Bristol
Bruges
Cambridge
Cardiff
Chester
Copenhagen
Edinburgh
Glasgow
Gothenburg
Hamburg

Barcelona
Benidorm
Cape_Town
Genoa
Granada
Helsinki
Innsbruck
Lisbon
Madrid
Marseille
Naples
Oslo
Rome
Saint_Petersb
San_Diego
San_Francisc

Berlin
Birmingham
Bratislava
Bregenz
Budapest
Dijon
Dresden
Dublin
Florence
Geneva
Ghent
Graz
Heidelberg
Inverness
Kraków
Linz

Bangkok
Fortaleza
Rio_de_Janeiro
Salvador,_Bahia

Beijing
Buenos_Aires
Hong_Kong
Istanbul
Moscow
New_York_City
São_Paulo
Seoul
Shanghai
Taipei
Tokyo

Mumbai
Varadero

Agra
Bahrain
Chengdu
Chennai
Chongqing
Dalian
Foz_do_Iguaçu
Goa
Guangzhou
Guilin
Hangzhou
Havana
Honolulu
Jerusalem
Kuala_Lumpur
Las_Vegas,_N

Atlanta
Boston
Cancún
Chicago
Florianópolis
Houston
Mexico_City
Miami
Montreal
Sydney

Los_Angeles
Toronto

Abu_Dhabi
Cairo
Dubai
Mecca

Figure 52: Clustering results for example 1

Some observations on these results:

- The cluster F contains two cities that lack on semantic information (the semantic attributes are missing), because this information has not been possible to obtained with the automatic extraction tools used. This corresponds to the cluster with Mumbai and Varadero.
- There is a well identified and compact cluster (J) with cities in deserts, with mosques, were people can dive:

	#3.Diving	#4.Mosque#2.Synagogue#2.Temple#2.Church	Desert
Abu_Dhabi	#Diving	#Mosque#Synagogue#Temple#Church	Desert
Cairo	?	#Mosque#Synagogue#Church#Abbey	Desert
Dubai	#Diving	#Mosque	Desert
Mecca	#Diving	#Mosque#Temple	Desert

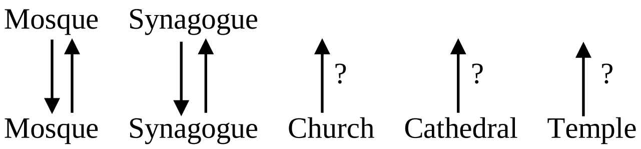
- There are four big clusters: A, B, C and G. It is interesting to realize that from the 150 (most touristic cities), the majority are European cities which usually are similar in terms of culture, what makes the cities also similar in buildings, popular sports, climate... In fact, 3 out of the 4 big clusters are for European cities. One of these clusters is based on Mediterranean climate and the other 2 have Oceanic climate.

- The 3 European clusters have very differentiated aquatic/nature sports:

Cluster A	Cluster B	Cluster C
#17.Swimming#21.Cycling#10.Sailing	#5.Swimming#5.Cycling#10.Sailing	?
#Swimming#Climbing#Cycling	#Swimming	?
#Cycling	?	?
#Sailing#Cycling	#Swimming#Surfing#Cycling#Climbing	#Rafting#Water_Polo#Kayaking
#Swimming#Cycling	?	?
#Surfing#Climbing#Skiing	?	?
#Swimming	#Swimming	?
#Cycling	#Skiing#Climbing	?
#Sailing#Cycling	#Sailing#Diving#Hunting	?
#Cycling	#Skiing	?
#Swimming#Sailing#Surfing#Skiing#...	#Swimming#Sailing#Windsurfing#Cycling	?
#Cycling	#Sailing#Water_Polo	?
#Swimming#Sailing#Cycling	#Swimming#Sailing#Skiing	?
#Swimming	#Water_Polo#Cycling	?
#Surfing#Cycling	#Sailing	?
#Diving#Swimming#Sailing#Water_Polo	#Sailing#Surfing#Cycling	#Swimming
#Sailing#Cycling	#Sailing#Windsurfing#Cycling	?
#Diving#Swimming#Cycling	?	?
#Sailing#Swimming#Cycling#Climbing	?	?
#Swimming	?	?
#Swimming#Cycling	#Sailing	?
#Swimming	#Sailing	?
#Swimming#Cycling	#Sailing	?
#Swimming#Cycling		?
#Cycling		?
#Sailing		?
#Sailing#Surfing#Skiing#Climbing#Cycling		?
#Swimming		?
#Skiing		?
#Kayaking#Swimming#Sailing#Cycling#...		?
#Swimming#Cycling#Climbing#Hunting		#Diving
#Cycling		?
		?

It is easy to see that in the first cluster predominate the concepts of Cycling and Swimming, there is also sailing, the concept that predominates in the second cluster, but at another level. The second centroid has the same concepts as the first one, but the weights are absolutely opposed. The third cluster is the one for the cities that have no information on aquatic/nature sports.

- One of the other features of the mechanism described in chapter 4 is that a city with a high different number of concepts in an attribute will have also a high distance. For instance, a city that all of its concepts appear in the cluster's centroid will have 0 distance, but the same is not equal the other way, since the centroid can have the same concepts as the city and so much more that the city does not have.



Of course, in the above example the concepts Church, Cathedral and Temple will be paired with Mosque or Synagogue and they will have a certain distance value, but even if this distance is low, it is above 0 and each of these concepts sum distance to the final distance between the cities. In other words, one can easily see the characteristic of the distance algorithm. The concepts shared by the city and the centroid add no distance, but the ones not shared does. With this, it can be said that the number of concepts in an attribute is determinant in the distance calculation and it affects the clustering.

It is interesting to analyse the same 3 European clusters for their water/geographical landmarks:

#Beach#River#Square#Hill#Terrace
#Canal#Lake#River#Bridge#Polder#Square#Terrace
#River#Polder#Bridge#Hill#Mountain
#River#Bridge#Beach#Stone_Bridge#Square#Hill#Terrace
#River#Bridge#Pedestrian_Bridge#Square#Hill#Mountain
#Beach#River
#Canal#River#Stone_Bridge#Bridge#Gorge#Square#Hill
#Canal#Beach#Bridge#Square
#River#Hill
#Canal#Lake#River#Beach#Hill#Mountain
#Canal#River#Bridge#Stone_Bridge#Hill
#Canal#Beach#Lake#Bridge#Square#Hill
#Canal#Bridge#Lake#River
…

#Beach
#Beach
#Beach#Cave#Square#Hill#Mountain
#Lake
#Square#Mountain
#River#Beach
#River#Bridge#Mountain
#River#Bridge#Square#Hill
#River#Lake#Square
#Canal#Beach#Cave#Square#Hill#Terrace#Mountain
#Cave#Square#Hill#Mountain
#Lake#River#Square#Hill#Mountain
#River#Bridge#Square#Hill
…

#Beach#River
#Canal#River
#Lake#River#Bridge
#Lake
#River#Bridge#Lake
?
#River#Bridge
#River#Bridge#Canal
#River#Bridge#Stone_Bridge
#Canal#Lake#River#Beach
#River#Bridge
#River#Bridge
#River#Bridge
…

The first cluster has cities with a large number of concepts, the second has an average number and the third has about two concepts per city. This is a pattern that is repeated in all the results. So, the clustering algorithm works well in this way.

The parameters used in this example are the ones that work with the idea of clustering of the DAMASK project. The weights for the numeric and categorical attributes are not high, but are enough to help to cluster the cities by these attributes. For example:

Cluster A				Numerical d.	Categorical d.
719.182,00	119,9529032	EU	Oceanic		
183.790,00	10,45	EU	Oceanic	0,254	0,000
741.636,00	12,66	EU	Oceanic	0,228	0,000
459.805,00	12,44	EU	Oceanic	0,233	0,000
93.238,00	22,61	EU	Oceanic	0,239	0,000
354.860,00	14,79	EU	Oceanic	0,234	0,000
139.001,00	26,49	EU	Oceanic	0,227	0,000
430.713,00	22,48	EU	Oceanic	0,214	0,000
116.709,00	13	EU	Oceanic	0,254	0,000
128.488,00	11,67	EU	Oceanic	0,256	0,000

Cluster B				Numerical d.	Categorical d.
1.115.309,00	105,1518182	EU	Mediterranean		
1.621.537,00	30,7	EU	Mediterranean	0,185	0,000
71.034,00	27,67	EU	Mediterranean	0,259	0,000
3.433.441,00	26,19	AF	Mediterranean	0,474	0,500
601.951,00	40,98	EU	Mediterranean	0,168	0,000
234.325,00	689,12	EU	Mediterranean	0,573	0,000
558.457,00	9,85	EU	Humid continental	0,228	0,500
112.467,00	570,43	EU	Subartic	0,580	0,500
517.802,00	40,33	EU	Mediterranean	0,179	0,000
3.255.944,00	649	EU	Mediterranean	0,684	0,000

Cluster D				Numerical d.	Categorical d.
4.895.059,00	179,975	SA	Tropical savanna		
5.104.476,00	11,92	AS	Tropical savanna	0,359	0,500
2.400.000,00	19,03	SA	Tropical savanna	0,587	0,000
6.023.699,00	5,68	SA	Tropical savanna	0,428	0,000
6.052.064,00	683,27	NA	Tropical savanna	0,591	0,500

Cluster E				Numerical d.	Categorical d.
9.862.533,00	107,6690909	AS	Humidsub-tropical		
7.480.601,00	50,09	AS	Humid continental	0,472	0,500
13.076.300,00	31,83	SA	Humidsub-tropical	0,635	0,500
7.012.738,00	4,6	AS	Humidsub-tropical	0,587	0,000
11.174.257,00	35,7	AS	Mediterranean	0,294	0,500
10.381.222,00	145,07	EU	Humid continental	0,127	1,000
8.175.133,00	9,9	NA	Humidsub-tropical	0,384	0,500
10.021.295,00	774,06	SA	Humidsub-tropical	0,549	0,500
10.349.312,00	68,64	AS	Humid continental	0,125	0,500
14.608.512,00	13,6	AS	Humidsub-tropical	0,860	0,000

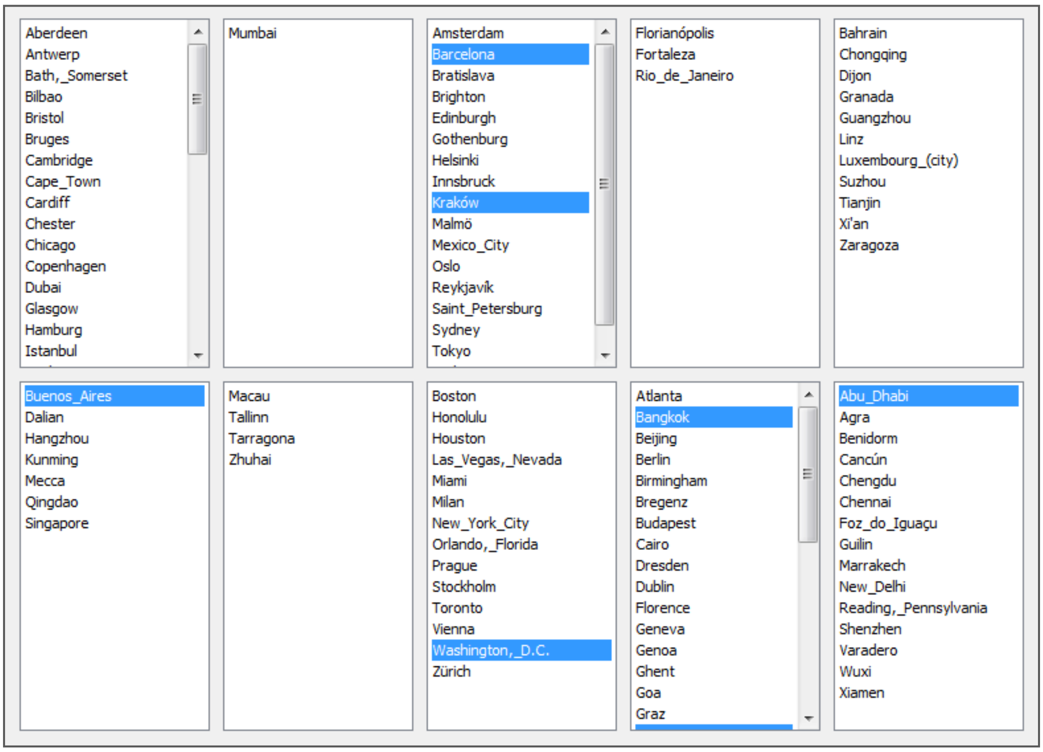
It can be seen that the cities are clustered for the numerical and categorical attributes. In this example, the last columns represent the numerical and categorical distances of each city to the centroid. It can be seen that the distances are low in at least one of the attributes types.

6.1.2 Semantic-only clustering

This section analyses what happens when the numerical and categorical weights are reduced to 0, in order to see the full behavior of the clustering algorithm for the semantic similarities.

These are the parameters for this example: OWA: Linguistic quantifiers (*most*), K = 10, λ = 0.3, N = 150, W_S= 1, W_N = 0, W_C = 0.

The 10 resulting clusters have these sizes: 43, 1, 17, 3, 11, 7, 4, 14, 35 and 15; see the results in figure 53.



Aberdeen
Antwerp
Bath,_Somerset
Bilbao
Bristol
Bruges
Cambridge
Cape_Town
Cardiff
Chester
Chicago
Copenhagen
Dubai
Glasgow
Hamburg
Istanbul

Mumbai

Amsterdam
Barcelona
Bratislava
Brighton
Edinburgh
Gothenburg
Helsinki
Innsbruck
Kraków
Malmö
Mexico_City
Oslo
Reykjavík
Saint_Petersburg
Sydney
Tokyo

Florianópolis
Fortaleza
Rio_de_Janeiro

Bahrain
Chongqing
Dijon
Granada
Guangzhou
Linz
Luxembourg_(city)
Suzhou
Tianjin
Xi'an
Zaragoza

Buenos_Aires
Dalian
Hangzhou
Kunming
Mecca
Qingdao
Singapore

Macau
Tallinn
Tarragona
Zhuhai

Boston
Honolulu
Houston
Las_Vegas,_Nevada
Miami
Milan
New_York_City
Orlando,_Florida
Prague
Stockholm
Toronto
Vienna
Washington,_D.C.
Zürich

Atlanta
Bangkok
Beijing
Berlin
Birmingham
Bregenz
Budapest
Cairo
Dresden
Dublin
Florence
Geneva
Genoa
Ghent
Goa
Graz

Abu_Dhabi
Agra
Benidorm
Cancún
Chengdu
Chennai
Foz_do_Iguaçu
Guilin
Marrakech
New_Delhi
Reading,_Pennsylvania
Shenzhen
Varadero
Wuxi
Xiamen

Figure 53: Clustering results for example 2.

Observations:

- Mumbai, one of the cities with no semantic information forms a cluster for itself.
- The cluster headed by Macau is a cluster with little semantic information. The semantic attributes of the cities of this cluster have just one or two concepts per attribute.

	?	#2.Lake#4.River	#4.Statue	#3.University
Macau	#Formula_One	#Lake#River	#Refuge#Statue	#University
Tallinn	#Ice_Hockey	#River	#Statue	#University#Library
Tarragona	?	#River	#Statue	#Roman_Amphitheatre
Zhuhai	#Motor_Sport	#Lake#River	#Park#Nature_Reserve#Statue	#University

- A cluster can be identified for the concepts that it has, but in some ways it can be also identified and interesting with regards to the concepts that it does not have. For instance, almost every city of the dataset has football and churches and museums. So, it is worth to know that the clustering method has generated a cluster for cities that do not have sports, nor the religious buildings (or just some general reference to that like *temple)* and also no museums:

	?	?	#6.Temple	?
Abu_Dhabi	#Diving	?	#Mosque#Synagogue#Temple#...	#Art_Gallery
Agra	?	?	#Mosque#Cathedral#Temple#...	?
Benidorm	?	?	?	?
Cancún	#Swimming	?	?	?
Chengdu	?	?	?	?
Chennai	?	#Cricket#Tennis	#Temple	?
Foz_do_Iguaçu	?	?	?	?
Guilin	?	?	?	?
Marrakech	?	?	#Mosque	?
New_Delhi	?	?	#Abbey	?
Reading	?	#Martial_Art#Golf	#Temple#Church	?
Shenzhen	?	?	?	?
Varadero	#Diving	?	?	?
Wuxi	?	?	#Temple	?
Xiamen	?	#Football	#Temple	?

- *Theater* is a concept that appears in almost every city for the *cultural buildings* attribute. But the clustering process has separated the cities with the theatre and the cities without it. For instance:

#6.School#8.University
#School#Library
#Public_University#Public_School#..
?
?
#University#School#Theater#..
#Theater#Music_School#University
#Library
#University#Theater#School
#University
?
?
#University#School#Library
?
#University#School#Theater
#University#Opera#Library

#10.University#6.School#6.Opera
#University
#University#School
#University
#University#School
#University#School#Opera#Private_School
#Private_University#School#University#..
#University#Music_School
#Opera#University
#University#School#Opera#Music_School
#Technological_University#School#Opera#..
#Opera#Forum

#16.Theater#10.School#13.Opera#17.University#8.Library
#Theater#School#Opera#University
#Theater#Opera#Forum#University#...
#University#Theater#Opera#Library#Business_School
#University#Theater#School
#University#Theater#Library#School#Opera
#University#Theater#Opera#Library#...
#University#Theater#School#Opera#...
#University#Theater#Opera
#University#Theater#School#Amphitheatre#Opera#...
#Theater#Opera#University
#University#Theater#School#Opera#Library#...
#University#Opera#Theater#School

#13.Theater#13.Opera#9.Library#6.Music_School#13.University#8.School
#Public_University#Theater#Public_School#University#...
#Public_University#Theater#Public_School#University#...
#University#Theater#School#Opera
#Opera#University
#Public_University#Theater#Public_School#...
#Theater#School#Opera#Library#Forum#University
#University#Theater#School#Opera#Music_School
#University#Theater#School#Business_School
#University#Theater#Opera#Library#Music_School
#Theater#Opera#Library#University#School
#University#Theater#School#Amphitheatre#Opera#...
#Private_University#Theater#School#Opera#University#...

#University#Library#Business_School	#University#Theater#Opera#Library#Music_School
#University#Theater#School#Opera#...	
#Theater#Opera#University#School	
#Library#Theater#University	
#University#Theater#Ancient_Greek_Theatre#...	

- We can also observe that when the cities are not clustered using numerical nor categorical values, the clusters do not show any homogeneity regarding these attributes, as expected:

167557	129,86	NA	Humidsub-tropical
617594	15,28	NA	Humidsub-tropical
371657	5,58	NA	Tropical savanna
2099451	12,05	NA	Humidsub-tropical
583756	608,33	NA	Desert
399457	1,67	NA	Tropical monsoon
1306661	122,17	EU	Humidsub-tropical
8175133	9,9	NA	Humidsub-tropical
238300	31,24	NA	Humidsub-tropical
1165581	191,09	EU	Oceanic
1253309	25,29	EU	Humid continental
4612191	166,84	NA	Humid continental
1691468	192,09	EU	Oceanic
601723	6,84	NA	Humidsub-tropical
341730	429,71	EU	Oceanic

1591748	217,9790909	AS	Humidsub-tropical
738004	4,96	AS	Desert
3967028	228,31	AS	Humidsub-tropical
149782	240,75	EU	Oceanic
234325	689,12	EU	Mediterranean
3152825	14,11	AS	Humidsub-tropical
181162	259,89	EU	Oceanic
76684	314,45	EU	Oceanic
1343091	13,57	AS	Humidsub-tropical
3766207	9,04	AS	Humid continental
3225812	413,48	AS	Humidsub-tropical
674317	210,09	EU	Oceanic

Only some few clusters present a majority regarding to a specific continent / climate. However, we believe that this fact can be caused by the natural similarity between cities with the same culture that are at the same time in the same continent and climate zone. For example:

1188010	123,956	AS	Humidsub-tropical
603492	6,03	AS	Desert
1430055	165,22	AS	Semi-arid
71034	27,67	EU	Mediterranean
542043	10,01	NA	Tropical savanna
3950437	499,63	AS	Humidsub-tropical
4328063	11,89	AS	Tropical savanna
293523	181,34	SA	Humidsub-tropical
649352	157,51	AS	Humidsub-tropical
839296	458,48	AF	Semi-arid
317797	212,07	AS	Humidsub-tropical
88082	92,05	NA	Humidsub-tropical
3000000	7,59	AS	Humidsub-tropical
20000	8,06	NA	Tropical savanna

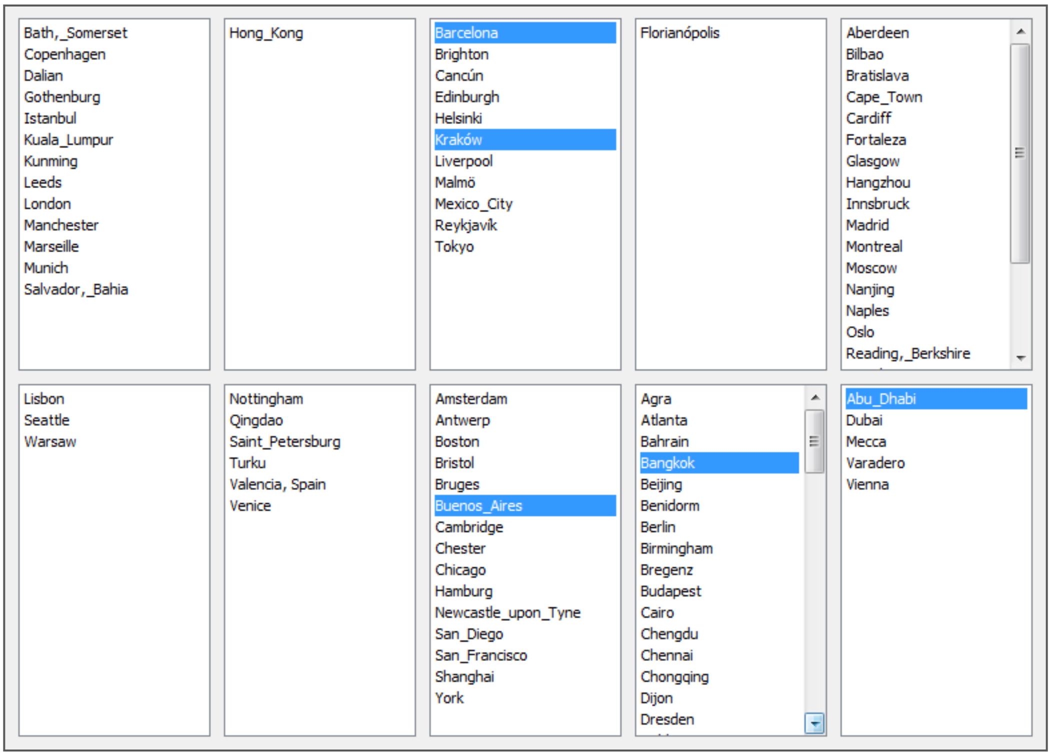
1108647	9,65	AS	Humidsub-tropical
578337	12,14	AS	Humidsub-tropical

6.1.3 One semantic attribute clustering

The idea behind this test example is to see what happens when the clustering is done considering just one semantic attribute. In the main code of the program, each semantic attribute has the same weight. For this test, all the weights were set to 0 except one. The attribute *aquatic/nature sports* was set to a weight of 1. No categorical nor numerical attribute is considered.

These are the parameters for this test: OWA: Linguistic quantifiers (*most*), K = 10, λ = 0.18, N = 150, W_S= 1, W_N = 0, W_C = 0.

The 10 resulting clusters have these sizes: 13, 1, 11, 1, 22, 3, 6, 15, 73 and 5; see the results in figure 54.



Bath,_Somerset
Copenhagen
Dalian
Gothenburg
Istanbul
Kuala_Lumpur
Kunming
Leeds
London
Manchester
Marseille
Munich
Salvador,_Bahia

Hong_Kong

Barcelona
Brighton
Cancún
Edinburgh
Helsinki
Kraków
Liverpool
Malmö
Mexico_City
Reykjavík
Tokyo

Florianópolis

Aberdeen
Bilbao
Bratislava
Cape_Town
Cardiff
Fortaleza
Glasgow
Hangzhou
Innsbruck
Madrid
Montreal
Moscow
Nanjing
Naples
Oslo
Reading,_Berkshire

Lisbon
Seattle
Warsaw

Nottingham
Qingdao
Saint_Petersburg
Turku
Valencia, Spain
Venice

Amsterdam
Antwerp
Boston
Bristol
Bruges
Buenos_Aires
Cambridge
Chester
Chicago
Hamburg
Newcastle_upon_Tyne
San_Diego
San_Francisco
Shanghai
York

Agra
Atlanta
Bahrain
Bangkok
Beijing
Benidorm
Berlin
Birmingham
Bregenz
Budapest
Cairo
Chengdu
Chennai
Chongqing
Dijon
Dresden

Abu_Dhabi
Dubai
Mecca
Varadero
Vienna

Figure 54: Clustering results for example 3

Observations:

- The cities with just one concept are clustered correctly for the mentioned concept:

	#11.Swimming
Barcelona	#Swimming
Brighton	#Swimming
Cancún	#Swimming
Edinburgh	#Swimming
Helsinki	#Swimming
Kraków	#Swimming
Liverpool	#Swimming
Malmö	#Swimming

	#5.Diving
Abu_Dhabi	#Diving
Dubai	#Diving
Mecca	#Diving
Varadero	#Diving
Vienna	#Diving

	#6.Sailing
Nottingham	#Sailing
Qingdao	#Sailing
Saint_Petersburg	#Sailing
Turku	#Sailing
Valencia, Spain	#Sailing
Venice	#Sailing

Mexico_City	#Swimming
Reykjavík	#Swimming
Tokyo	#Swimming

- The cluster with 73 cities is the one in which the cities do not have any *aquatic/nature sport*.
- The other clusters are combinations of different values for the sports concepts. For instance:

	#13.Swimming#12.Cycling#6.Sailing#3.Diving
Bath,_Somerset	#Swimming#Cycling
Copenhagen	#Swimming#Sailing#Cycling
Dalian	#Swimming#Cycling
Gothenburg	#Diving#Swimming#Sailing#Water_Polo
Istanbul	#Swimming#Sailing#Cycling#Mountain_Biking
Kuala_Lumpur	#Swimming#Cycling
Kunming	#Diving#Swimming#Cycling
Leeds	#Sailing#Swimming#Cycling#Climbing
London	#Swimming#Cycling
Manchester	#Swimming#Cycling
Marseille	#Swimming#Sailing#Windsurfing#Cycling
Munich	#Swimming#Cycling
Salvador,_Bahia	#Diving#Swimming#Sailing#Cycling#Hunting

- There are two clusters that only have one city. This is because they show quite a unique profile. Whereas in most of the cities it is possible to swim and cycle, in Hong Kong we do not have the cycling possibility. In Florianópolis the combination of Snowboarding with Diving and Surfing is quite uniqu. These are the clusters:

	#1.Swimming#1.Sailing
Hong_Kong	#Swimming#Sailing

	#1.Diving#1.Surfing#1.Cycling#1.Mountain_Biking#1.Snowboarding
Florianópolis	#Diving#Surfing#Cycling#Mountain_Biking#Snowboarding

Here is another example of why a city like Copenhagen, which is similar to Hong Kong, do not cluster with it:

Copenhagen	#Swimming#Sailing#Cycling

The only difference is that Copenhagen has *cycling* and Hong Kong does not. It is a priori a small difference, but there is a centroid that is also similar to the concepts offered by Copenhagen:

#13.Swimming#12.Cycling#6.Sailing#3.Diving

Here the difference is just the *diving* concept. So, it can be said that the reason for Copenhagen to belong to the cluster represented by the above centroid is that the lack of the low weighted concept *diving* has minor influence than the concept of excess that is *cycling*.

6.1.4 Variation on the lambda cut value and variation of K

This test continues to consider only the semantic information, which permits to better interpret the reasons for grouping the cities.

This test is done to show how the system responds to a variation on the number of clusters. Also, a small change on the lambda cut mechanism, which is used to limit the number of concepts that a centroid attribute will have, is done. Remember the cut formula:

$$c = \{\langle n_i, t_i \rangle | n_i > max(n * \lambda, 1)\},$$

The centroid will only have those concepts that appear in a certain percentage of the cities that the cluster has.

These are the parameters for this example: OWA: Linguistic quantifiers (*most*), K = 12, λ = 0.3, N = 150, W_S= 1, W_N = 0, W_C = 0.

Using these parameters, the execution of the test unveiled a problem when there are various clusters with low quantity of cities, such as the following example:

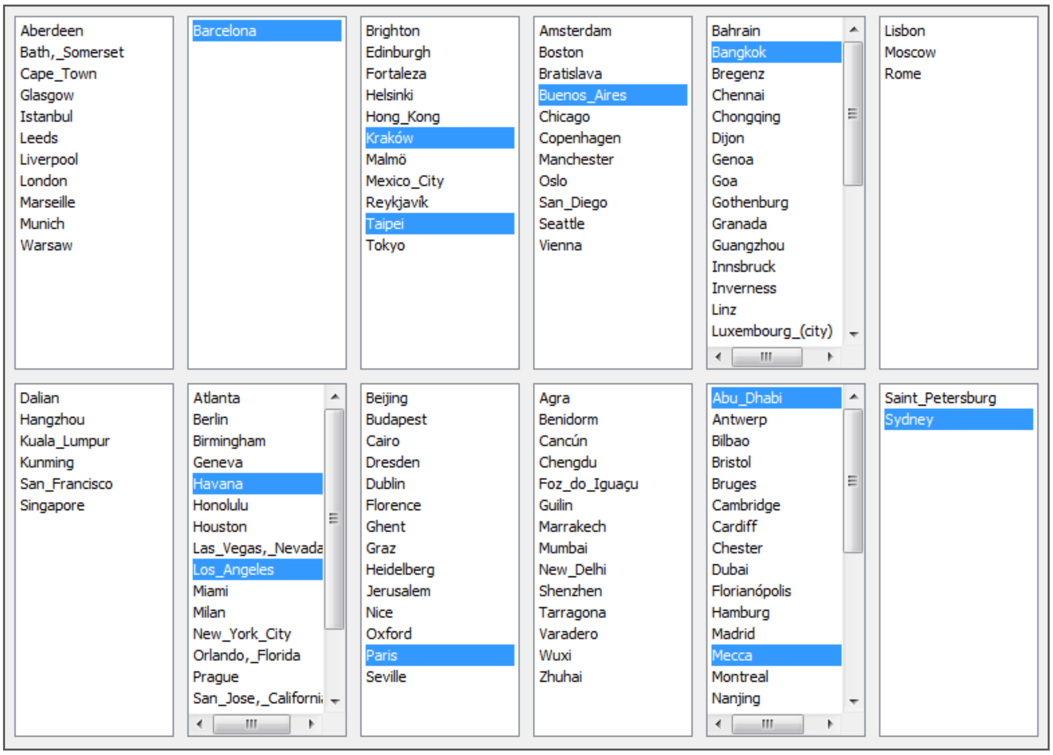
	?	?	?
Agra	?	?	#Mosque#Cathedral#Temple#Sanctuary
Benidorm	?	?	?
Cancún	#Swimming	?	?
Chengdu	?	?	?
Foz_do_Iguaçu	?	?	?
Guilin	?	?	?
Marrakech	?	?	#Mosque
Mumbai	?	?	?
New_Delhi	?	?	#Abbey
Shenzhen	?	?	?
Tarragona	?	?	?
Varadero	#Diving	?	?
Wuxi	?	?	#Temple
Zhuhai	?	#Motor_Sport	?

In this case, for *n* = 14 and λ = 0.3, the minimum frequency required to each concept is 14 x 0.3 = 4.2. Therefore, only the terms that can be found in 5 or more cities will be displayed in the centroid. Since the description of those cities is almost empty, the centroid generated is also empty (marked with *?*). What can happen here is that at the next step, all the cities of these clusters are moved to another cluster, leaving this cluster empty. In the K-means algorithm, a cluster cannot be empty. In order to fix this problem, the formula to cut the centroids was changed to the following one (just for this test):

$$c = \{\langle n_i, t_i \rangle | n_i > 1\},$$

This workaround is experimental and should be used just in the case where the aforementioned problem appears, because then the centroid is not restricting the frequency.

With this modification, the 12 resulting clusters have these sizes: 11, 1, 11, 11, 29, 3, 6, 11, 19, 14, 14 and 29; see the results in figure 55.



Aberdeen
Bath,_Somerset
Cape_Town
Glasgow
Istanbul
Leeds
Liverpool
London
Marseille
Munich
Warsaw

Barcelona

Brighton
Edinburgh
Fortaleza
Helsinki
Hong_Kong
Kraków
Malmö
Mexico_City
Reykjavik
Taipei
Tokyo

Amsterdam
Boston
Bratislava
Buenos_Aires
Chicago
Copenhagen
Manchester
Oslo
San_Diego
Seattle
Vienna

Bahrain
Bangkok
Bregenz
Chennai
Chongqing
Dijon
Genoa
Goa
Gothenburg
Granada
Guangzhou
Innsbruck
Inverness
Linz
Luxembourg_(city)

Lisbon
Moscow
Rome

Dalian
Hangzhou
Kuala_Lumpur
Kunming
San_Francisco
Singapore

Atlanta
Berlin
Birmingham
Geneva
Havana
Honolulu
Houston
Las_Vegas,_Nevada
Los_Angeles
Miami
Milan
New_York_City
Orlando,_Florida
Prague
San_Jose,_Californi

Beijing
Budapest
Cairo
Dresden
Dublin
Florence
Ghent
Graz
Heidelberg
Jerusalem
Nice
Oxford
Paris
Seville

Agra
Benidorm
Cancún
Chengdu
Foz_do_Iguaçu
Guilin
Marrakech
Mumbai
New_Delhi
Shenzhen
Tarragona
Varadero
Wuxi
Zhuhai

Abu_Dhabi
Antwerp
Bilbao
Bristol
Bruges
Cambridge
Cardiff
Chester
Dubai
Florianópolis
Hamburg
Madrid
Mecca
Montreal
Nanjing

Saint_Petersburg
Sydney

Figure 55: Clustering results for example 3

Observations when increasing the number of clusters and lambda threshold:

- Some interesting results appear in the *water/geographical landmarks*. For instance, there is a perfectly defined cluster:

	#3.River#3.Bridge#3.Square#3.Hill
Lisbon	#River#Bridge#Square#Hill
Moscow	#River#Bridge#Square#Hill
Rome	#River#Bridge#Square#Hill

And another example:

	#14.River#11.Bridge
Beijing	#River#Stone_Bridge#Pedestrian_Bridge
Budapest	#River#Bridge#Lake
Cairo	#River#Beach#Bridge
Dresden	#River#Bridge
Dublin	#River#Bridge#Canal
Florence	#River#Bridge#Stone_Bridge
Ghent	#River#Bridge
Graz	#River#Bridge
Heidelberg	#River#Bridge
Jerusalem	#River
Nice	#Beach#River#Bridge
Oxford	#Canal#River#Bridge
Paris	#Canal#River#Bridge
Seville	#River

- The *aquatic/nature sports* also provides interesting results. The majority of cities with no information on this attribute are clustered together but divided in various clusters, while the cities with information are in other clusters. For example:

	#9.Swimming
Brighton	#Swimming
Edinburgh	#Swimming
Fortaleza	#Windsurfing#Surfing
Helsinki	#Swimming
Hong_Kong	#Swimming#Sailing
Kraków	#Swimming
Malmö	#Swimming
Mexico_City	#Swimming
Reykjavík	#Swimming
Taipei	?
Tokyo	#Swimming

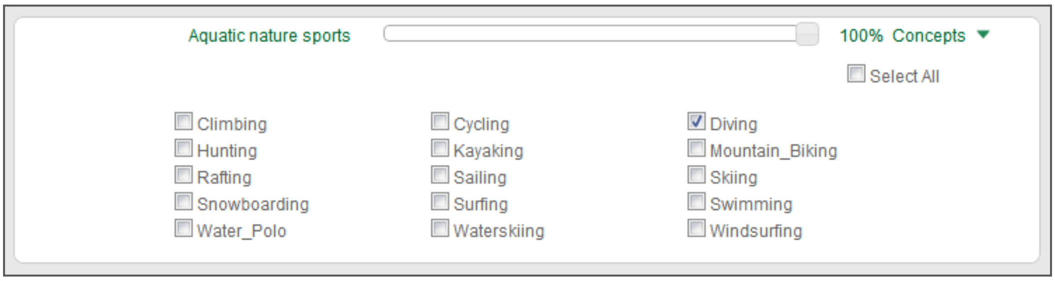
	#13.Sailing#12.Cycling
Abu_Dhabi	#Diving
Antwerp	#Sailing#Cycling
Bilbao	#Surfing#Climbing#Skiing
Bristol	#Cycling
Bruges	#Sailing#Cycling
Cambridge	#Cycling
Cardiff	#Swimming#Sailing#Surfing#Skiing#...
Chester	#Cycling
Dubai	#Diving
Florianópolis	#Diving#Surfing#Cycling#...
Hamburg	#Sailing#Cycling
Madrid	#Skiing
Mecca	#Diving
Montreal	#Skiing
Nanjing	#Climbing
Naples	#Sailing#Water_Polo
Newcastle_upon_Tyne	#Cycling
Nottingham	#Sailing
Qingdao	#Sailing

6.2 Study of the Recommendation System

In the previous section, the clustering algorithm was studied at detail. This section will study the behavior of the Web recommender. Considering that the clusters are already created with the parameters and results from the first test shown above, this study concentrated on the results that the recommender system returns for a given user's profile, which works here as the parameter component.

6.2.1 Easy matching

The idea behind this test is to simplify the user profile to the maximum (to set just one semantic attribute on the profile page) and try to get the cluster that matches the requested concepts. For instance, the cluster headed by Abu Dhabi has just one concept in its *aquatic/nature sports* attribute, which is "*diving"*. If a user checks the diving concept and sets *aquatic/nature sports* to 100% and all the other attributes to 0%, the recommendation expected is the aforementioned cluster.



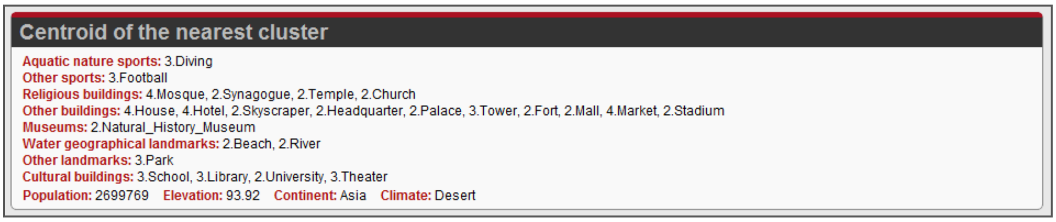
Aquatic nature sports 100% Concepts ▼

□ Select All

□ Climbing □ Cycling ☑ Diving
□ Hunting □ Kayaking □ Mountain_Biking
□ Rafting □ Sailing □ Skiing
□ Snowboarding □ Surfing □ Swimming
□ Water_Polo □ Waterskiing □ Windsurfing

Figure 56: User profile with aquatic/nature sports at 100% and diving checked.

The recommended cluster's centroid is shown in figure 57:



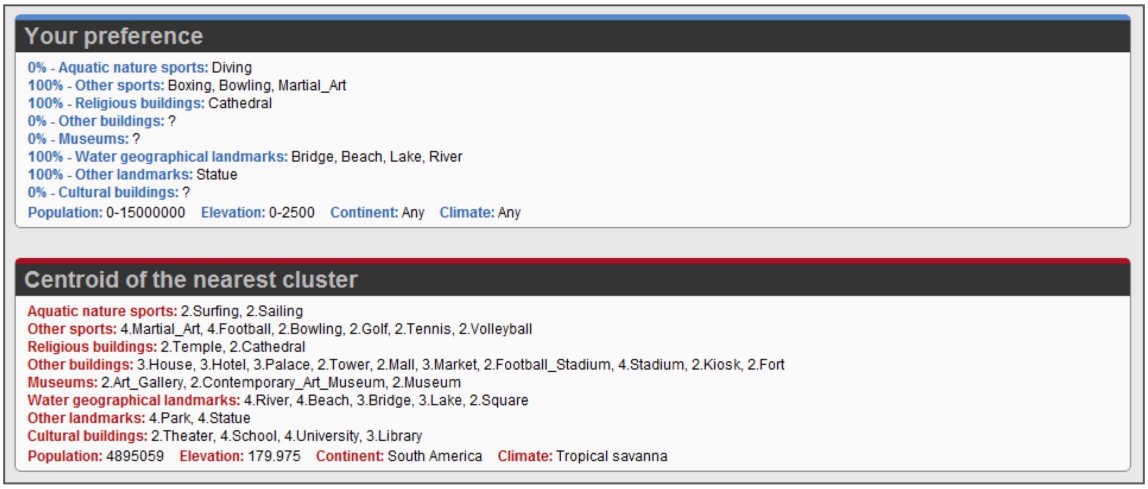
Centroid of the nearest cluster

Aquatic nature sports: 3.Diving
Other sports: 3.Football
Religious buildings: 4.Mosque, 2.Synagogue, 2.Temple, 2.Church
Other buildings: 4.House, 4.Hotel, 2.Skyscraper, 2.Headquarter, 2.Palace, 3.Tower, 2.Fort, 2.Mall, 4.Market, 2.Stadium
Museums: 2.Natural_History_Museum
Water geographical landmarks: 2.Beach, 2.River
Other landmarks: 3.Park
Cultural buildings: 3.School, 3.Library, 2.University, 3.Theater
Population: 2699769 Elevation: 93.92 Continent: Asia Climate: Desert

Figure 57: Centroid of the recommended cluster for example 1.

6.2.2 Profile with preference values similar to a centroid

It is expected that a profile with values that are similar to the centroid of some cluster, results in a recommendation for this particular cluster. For this test, some attributes of the user's profile have been set to 100% and some of the concepts for them were checked in a way that the resulting prototype will be similar to the expected recommended centroid. Figure 58 shows the user's profile and the resulting recommendation, which is the expected one:



Your preference

0% - Aquatic nature sports: Diving
100% - Other sports: Boxing, Bowling, Martial_Art
100% - Religious buildings: Cathedral
0% - Other buildings: ?
0% - Museums: ?
100% - Water geographical landmarks: Bridge, Beach, Lake, River
100% - Other landmarks: Statue
0% - Cultural buildings: ?
Population: 0-15000000 Elevation: 0-2500 Continent: Any Climate: Any

Centroid of the nearest cluster

Aquatic nature sports: 2.Surfing, 2.Sailing
Other sports: 4.Martial_Art, 4.Football, 2.Bowling, 2.Golf, 2.Tennis, 2.Volleyball
Religious buildings: 2.Temple, 2.Cathedral
Other buildings: 3.House, 3.Hotel, 3.Palace, 2.Tower, 2.Mall, 3.Market, 2.Football_Stadium, 4.Stadium, 2.Kiosk, 2.Fort
Museums: 2.Art_Gallery, 2.Contemporary_Art_Museum, 2.Museum
Water geographical landmarks: 4.River, 4.Beach, 3.Bridge, 3.Lake, 2.Square
Other landmarks: 4.Park, 4.Statue
Cultural buildings: 2.Theater, 4.School, 4.University, 3.Library
Population: 4895059 Elevation: 179.975 Continent: South America Climate: Tropical savanna

Figure 58: Prototype and recommended cluster's centroid for example 2.

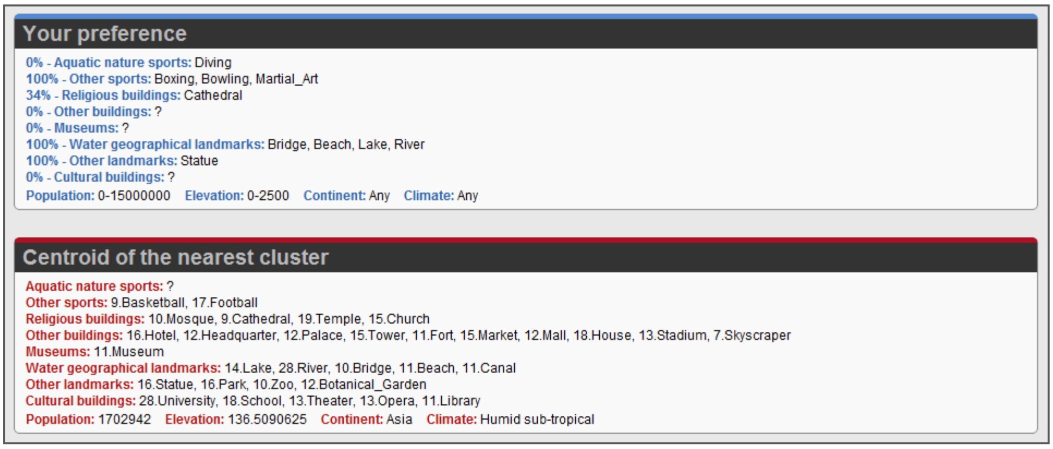
Notice that the prototype is far from being equal to the centroid. For instance, the *other sports* attribute is just equal for the *martial art* and *bowling* attributes. The centroid is far from that, although the *martial art* concept has bigger weight than others. For *religious buildings* only half of the concepts are directly equal. The semantic similarity in *Water/geographical* is high since there is just to one concept different, and it is the one with lowest weight. And finally, the *other landmarks* attribute only has the half terms similar.

This example demonstrates that without having an exact prototype-centroid match the system is able to retrieve the correct cluster.

6.2.3 Variation on the attribute strengths

This test is a variation of the previous one where the only thing modified are strengths applied to the attributes in order to see if this can make vary the recommendation.

The strength on the religious buildings has been reduced to 34%. We will analyze if small changes like the one applied here can be enough to make the recommendation system changes its results. And that is the case shown in figure 59:



Your preference

0% - Aquatic nature sports: Diving
100% - Other sports: Boxing, Bowling, Martial_Art
34% - Religious buildings: Cathedral
0% - Other buildings: ?
0% - Museums: ?
100% - Water geographical landmarks: Bridge, Beach, Lake, River
100% - Other landmarks: Statue
0% - Cultural buildings: ?
Population: 0-15000000 Elevation: 0-2500 Continent: Any Climate: Any

Centroid of the nearest cluster

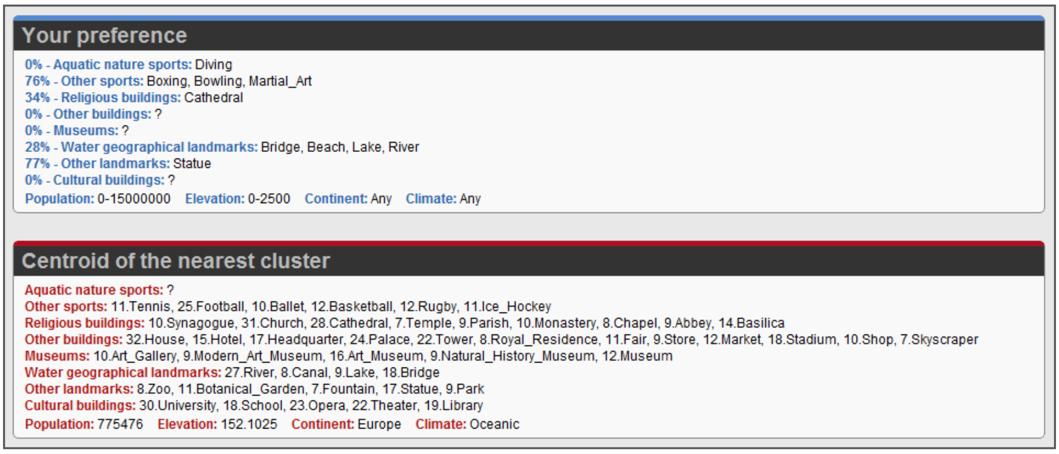
Aquatic nature sports: ?
Other sports: 9.Basketball, 17.Football
Religious buildings: 10.Mosque, 9.Cathedral, 19.Temple, 15.Church
Other buildings: 16.Hotel, 12.Headquarter, 12.Palace, 15.Tower, 11.Fort, 15.Market, 12.Mall, 18.House, 13.Stadium, 7.Skyscraper
Museums: 11.Museum
Water geographical landmarks: 14.Lake, 28.River, 10.Bridge, 11.Beach, 11.Canal
Other landmarks: 16.Statue, 16.Park, 10.Zoo, 12.Botanical_Garden
Cultural buildings: 28.University, 18.School, 13.Theater, 13.Opera, 11.Library
Population: 1702942 Elevation: 136.5090625 Continent: Asia Climate: Humid sub-tropical

Figure 59: Prototype and recommended cluster's centroid for example 3.

Now the resulting centroid does not seem that similar to the prototype at first sight, if we consider them globally. However, if we concentrate in the *religious buildings* attribute, the user searches cities with *cathedrals* but with less intensity than before, so the system has selected a group of cities that have a lot of religions buildings, including also other kinds of temples and churches. So, the system has concentrated more on finding catholic religious buildings in a more general view. Other attributes have maximized their similarity, like the *water/geographical landmarks* attribute and the *other landmarks* attribute has the match for *statue* with a high weight. On the contrary, the *other sports* concept is now absolutely different.

It is easy to see that the result is still good enough, provided that the cluster recommended in the previous text would have been also a good recommendation.

Another interesting result happens when varying even more the strengths of preference of the attributes. In figure 60 can be seen that the strengths for some attributes have been lowed. The recommendation result is again another different cluster. And notice that in the whole example, no concept has been touched.



Your preference

0% - Aquatic nature sports: Diving
76% - Other sports: Boxing, Bowling, Martial_Art
34% - Religious buildings: Cathedral
0% - Other buildings: ?
0% - Museums: ?
28% - Water geographical landmarks: Bridge, Beach, Lake, River
77% - Other landmarks: Statue
0% - Cultural buildings: ?
Population: 0-15000000 Elevation: 0-2500 Continent: Any Climate: Any

Centroid of the nearest cluster

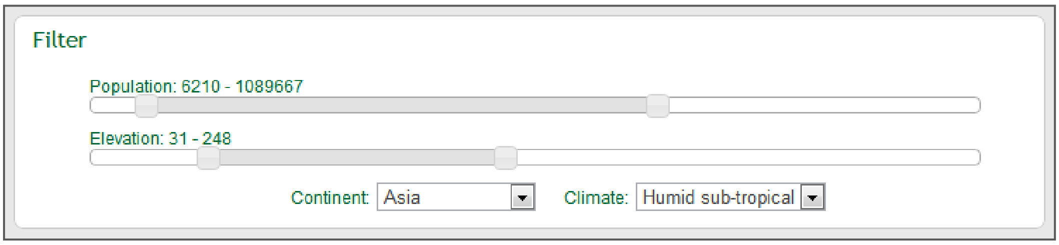
Aquatic nature sports: ?
Other sports: 11.Tennis, 25.Football, 10.Ballet, 12.Basketball, 12.Rugby, 11.Ice_Hockey
Religious buildings: 10.Synagogue, 31.Church, 28.Cathedral, 7.Temple, 9.Parish, 10.Monastery, 8.Chapel, 9.Abbey, 14.Basilica
Other buildings: 32.House, 15.Hotel, 17.Headquarter, 24.Palace, 22.Tower, 8.Royal_Residence, 11.Fair, 9.Store, 12.Market, 18.Stadium, 10.Shop, 7.Skyscraper
Museums: 10.Art_Gallery, 9.Modern_Art_Museum, 16.Art_Museum, 9.Natural_History_Museum, 12.Museum
Water geographical landmarks: 27.River, 8.Canal, 9.Lake, 18.Bridge
Other landmarks: 8.Zoo, 11.Botanical_Garden, 7.Fountain, 17.Statue, 9.Park
Cultural buildings: 30.University, 18.School, 23.Opera, 22.Theater, 19.Library
Population: 775476 Elevation: 152.1025 Continent: Europe Climate: Oceanic

Figure 60: Prototype and recommended cluster's centroid for example 3b.

We can conclude that the strength or importance given to the attributes is a sensitive value in the system.

6.2.4 Filtering the recommendation

There are several clusters that are quite big (around 30 cities), so that the user will receive a too long list of alternatives, which may confuse more than help with the recommendation. In those cases, the user has the possibility to use the filtering tool in his profile, as detailed in chapter 5 and shown in figure 61.



Filter

Population: 6210 - 1089667

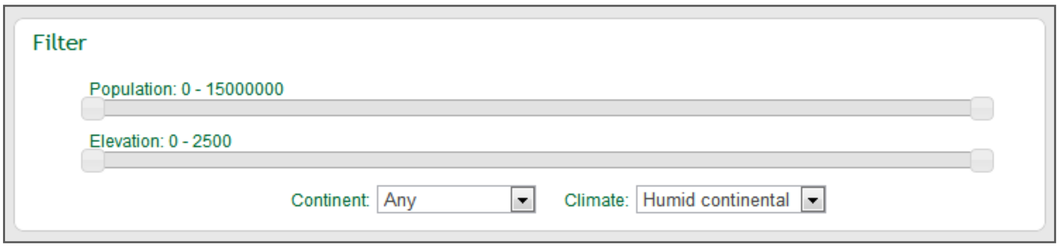
Elevation: 31 - 248

Continent: Asia

Climate: Humid sub-tropical

Figure 61: Filter at user's profile.

From the last results of the previous test, a filtering has been done to show only cities that have a *humid continental* climate. The rest of the filter is left as default:



Filter

Population: 0 - 15000000

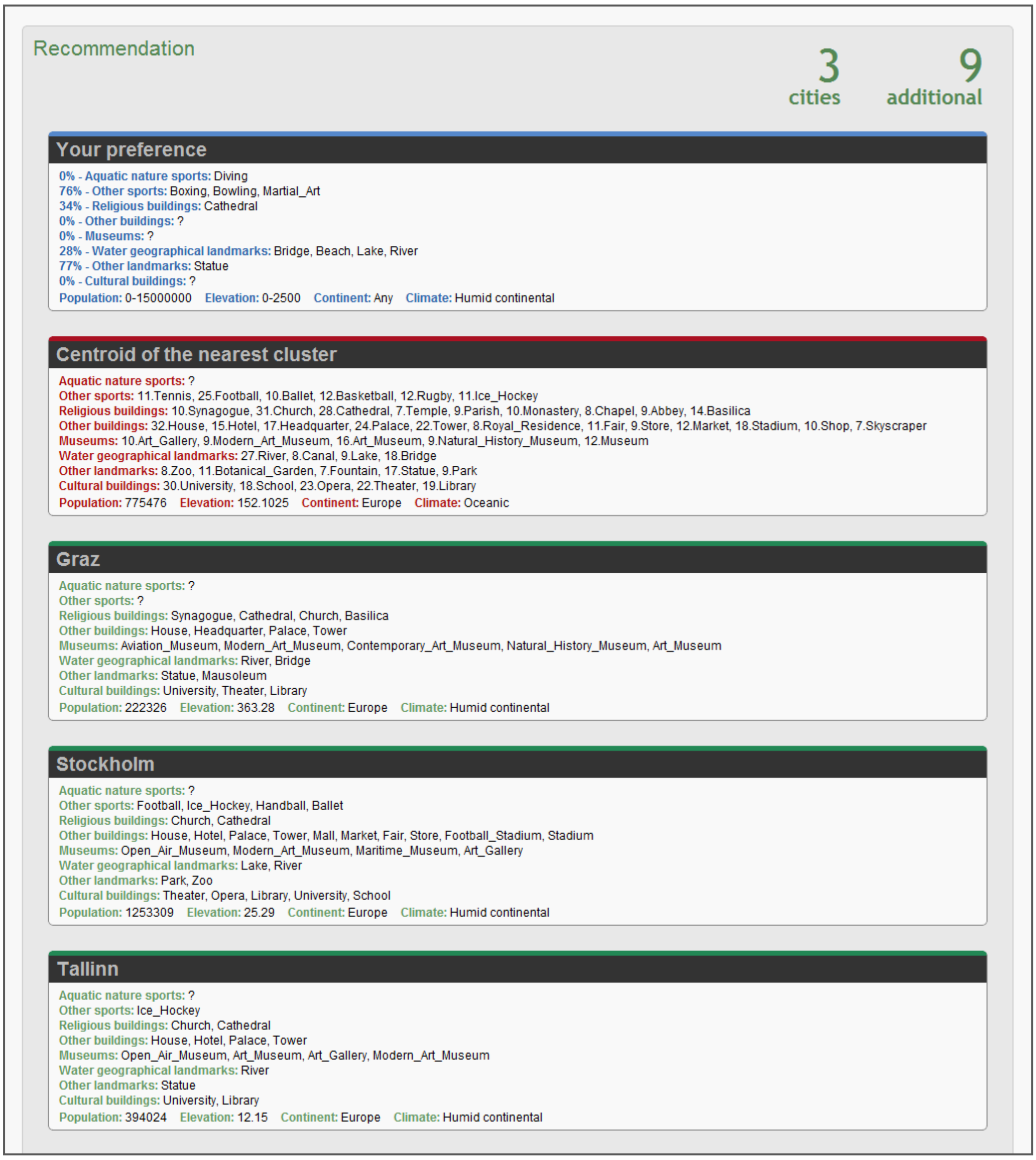
Elevation: 0 - 2500

Continent: Any

Climate: Humid continental

Figure 62: Filter applied for the example.

As the majority of the cities of this cluster, which has 32 cities, have an *oceanic* climate, the result only leaves 3 cities, the 3 that match the filter requirements. These 3 cities are shown in figure 63:



Recommendation

3 cities 9 additional

Your preference

0% - Aquatic nature sports: Diving
76% - Other sports: Boxing, Bowling, Martial_Art
34% - Religious buildings: Cathedral
0% - Other buildings: ?
0% - Museums: ?
28% - Water geographical landmarks: Bridge, Beach, Lake, River
77% - Other landmarks: Statue
0% - Cultural buildings: ?
Population: 0-15000000 Elevation: 0-2500 Continent: Any Climate: Humid continental

Centroid of the nearest cluster

Aquatic nature sports: ?
Other sports: 11.Tennis, 25.Football, 10.Ballet, 12.Basketball, 12.Rugby, 11.Ice_Hockey
Religious buildings: 10.Synagogue, 31.Church, 28.Cathedral, 7.Temple, 9.Parish, 10.Monastery, 8.Chapel, 9.Abbey, 14.Basilica
Other buildings: 32.House, 15.Hotel, 17.Headquarter, 24.Palace, 22.Tower, 8.Royal_Residence, 11.Fair, 9.Store, 12.Market, 18.Stadium, 10.Shop, 7.Skyscraper
Museums: 10.Art_Gallery, 9.Modern_Art_Museum, 16.Art_Museum, 9.Natural_History_Museum, 12.Museum
Water geographical landmarks: 27.River, 8.Canal, 9.Lake, 18.Bridge
Other landmarks: 8.Zoo, 11.Botanical_Garden, 7.Fountain, 17.Statue, 9.Park
Cultural buildings: 30.University, 18.School, 23.Opera, 22.Theater, 19.Library
Population: 775476 Elevation: 152.1025 Continent: Europe Climate: Oceanic

Graz

Aquatic nature sports: ?
Other sports: ?
Religious buildings: Synagogue, Cathedral, Church, Basilica
Other buildings: House, Headquarter, Palace, Tower
Museums: Aviation_Museum, Modern_Art_Museum, Contemporary_Art_Museum, Natural_History_Museum, Art_Museum
Water geographical landmarks: River, Bridge
Other landmarks: Statue, Mausoleum
Cultural buildings: University, Theater, Library
Population: 222326 Elevation: 363.28 Continent: Europe Climate: Humid continental

Stockholm

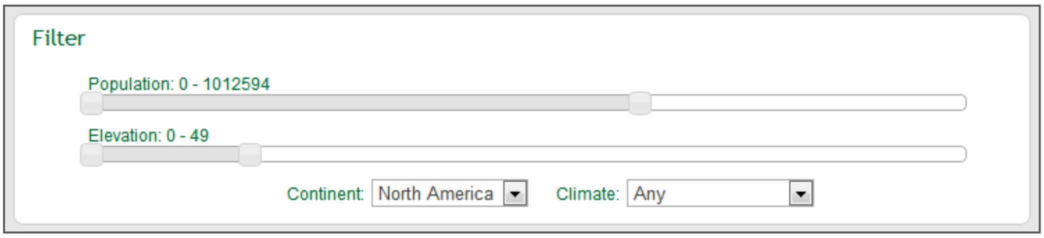
Aquatic nature sports: ?
Other sports: Football, Ice_Hockey, Handball, Ballet
Religious buildings: Church, Cathedral
Other buildings: House, Hotel, Palace, Tower, Mall, Market, Fair, Store, Football_Stadium, Stadium
Museums: Open_Air_Museum, Modern_Art_Museum, Maritime_Museum, Art_Gallery
Water geographical landmarks: Lake, River
Other landmarks: Park, Zoo
Cultural buildings: Theater, Opera, Library, University, School
Population: 1253309 Elevation: 25.29 Continent: Europe Climate: Humid continental

Tallinn

Aquatic nature sports: ?
Other sports: Ice_Hockey
Religious buildings: Church, Cathedral
Other buildings: House, Hotel, Palace, Tower
Museums: Open_Air_Museum, Art_Museum, Art_Gallery, Modern_Art_Museum
Water geographical landmarks: River
Other landmarks: Statue
Cultural buildings: University, Library
Population: 394024 Elevation: 12.15 Continent: Europe Climate: Humid continental

Figure 63: Filtered recommendation for example 4.

The following is a more complex filtering example. The process recommends a cluster with 10 cities. Suppose that a user wants the tiniest North American cities located in the coast. So, the user would set a filter like the one shown in figure 64:



Filter

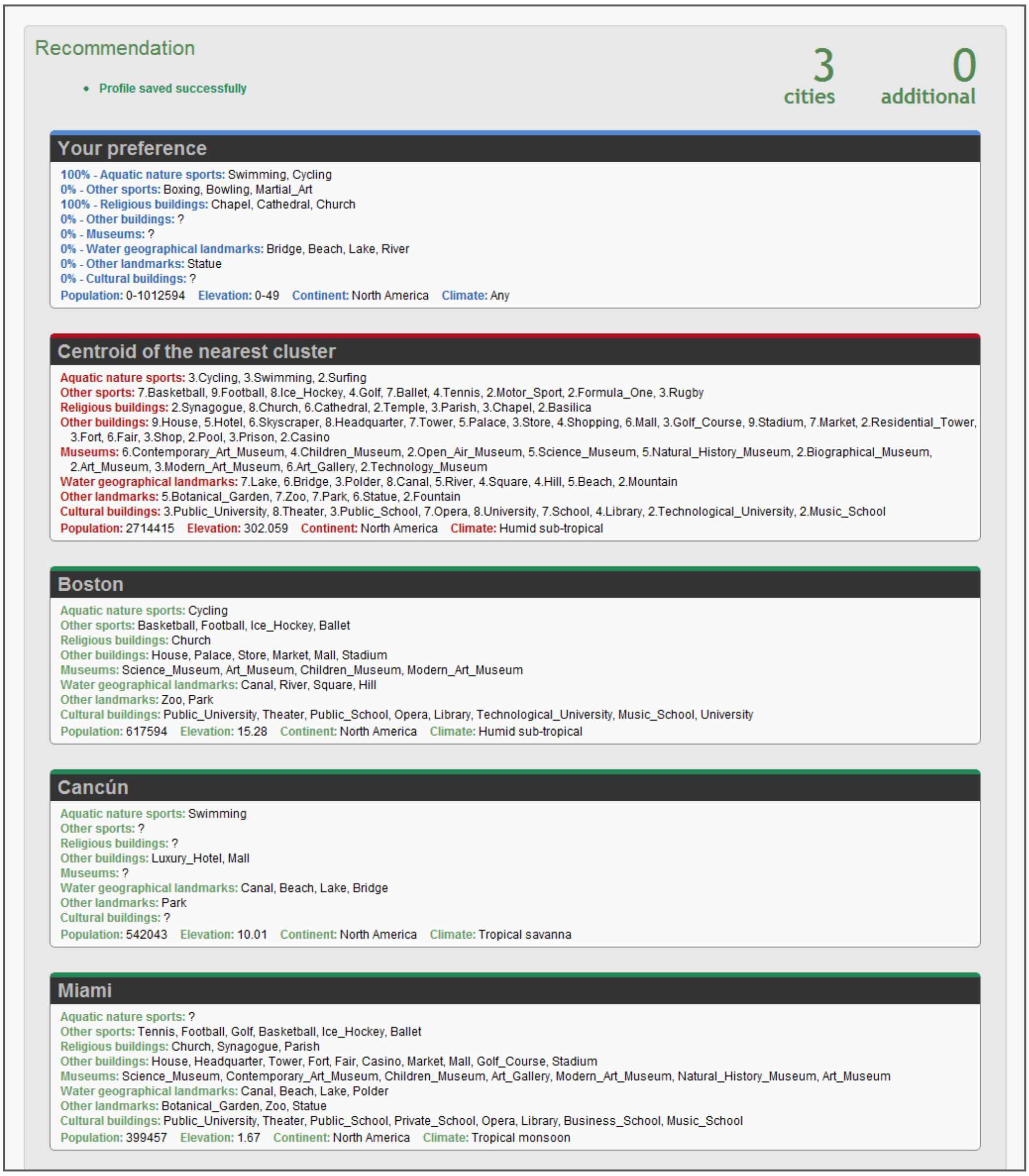
Population: 0 - 1012594

Elevation: 0 - 49

Continent: North America Climate: Any

Figure 64: Filter for example 4b.

The cluster to filter is formed by the following cities: Atlanta, Boston, Cancun, Chicago, Florianopolis, Houston, Mexico City, Miami, Montreal and Sydney. After applying the filtering, this is the result:



Recommendation

- Profile saved successfully

3 cities 0 additional

Your preference

100% - Aquatic nature sports: Swimming, Cycling
0% - Other sports: Boxing, Bowling, Martial_Art
100% - Religious buildings: Chapel, Cathedral, Church
0% - Other buildings: ?
0% - Museums: ?
0% - Water geographical landmarks: Bridge, Beach, Lake, River
0% - Other landmarks: Statue
0% - Cultural buildings: ?
Population: 0-1012594 Elevation: 0-49 Continent: North America Climate: Any

Centroid of the nearest cluster

Aquatic nature sports: 3.Cycling, 3.Swimming, 2.Surfing
Other sports: 7.Basketball, 9.Football, 8.Ice_Hockey, 4.Golf, 7.Ballet, 4.Tennis, 2.Motor_Sport, 2.Formula_One, 3.Rugby
Religious buildings: 2.Synagogue, 8.Church, 6.Cathedral, 2.Temple, 3.Parish, 3.Chapel, 2.Basilica
Other buildings: 9.House, 5.Hotel, 6.Skyscraper, 8.Headquarter, 7.Tower, 5.Palace, 3.Store, 4.Shopping, 6.Mall, 3.Golf_Course, 9.Stadium, 7.Market, 2.Residential_Tower, 3.Fort, 6.Fair, 3.Shop, 2.Pool, 3.Prison, 2.Casino
Museums: 6.Contemporary_Art_Museum, 4.Children_Museum, 2.Open_Air_Museum, 5.Science_Museum, 5.Natural_History_Museum, 2.Biographical_Museum, 2.Art_Museum, 3.Modern_Art_Museum, 6.Art_Gallery, 2.Technology_Museum
Water geographical landmarks: 7.Lake, 6.Bridge, 3.Polder, 8.Canal, 5.River, 4.Square, 4.Hill, 5.Beach, 2.Mountain
Other landmarks: 5.Botanical_Garden, 7.Zoo, 7.Park, 6.Statue, 2.Fountain
Cultural buildings: 3.Public_University, 8.Theater, 3.Public_School, 7.Opera, 8.University, 7.School, 4.Library, 2.Technological_University, 2.Music_School
Population: 2714415 Elevation: 302.059 Continent: North America Climate: Humid sub-tropical

Boston

Aquatic nature sports: Cycling
Other sports: Basketball, Football, Ice_Hockey, Ballet
Religious buildings: Church
Other buildings: House, Palace, Store, Market, Mall, Stadium
Museums: Science_Museum, Art_Museum, Children_Museum, Modern_Art_Museum
Water geographical landmarks: Canal, River, Square, Hill
Other landmarks: Zoo, Park
Cultural buildings: Public_University, Theater, Public_School, Opera, Library, Technological_University, Music_School, University
Population: 617594 Elevation: 15.28 Continent: North America Climate: Humid sub-tropical

Cancún

Aquatic nature sports: Swimming
Other sports: ?
Religious buildings: ?
Other buildings: Luxury_Hotel, Mall
Museums: ?
Water geographical landmarks: Canal, Beach, Lake, Bridge
Other landmarks: Park
Cultural buildings: ?
Population: 542043 Elevation: 10.01 Continent: North America Climate: Tropical savanna

Miami

Aquatic nature sports: ?
Other sports: Tennis, Football, Golf, Basketball, Ice_Hockey, Ballet
Religious buildings: Church, Synagogue, Parish
Other buildings: House, Headquarter, Tower, Fort, Fair, Casino, Market, Mall, Golf_Course, Stadium
Museums: Science_Museum, Contemporary_Art_Museum, Children_Museum, Art_Gallery, Modern_Art_Museum, Natural_History_Museum, Art_Museum
Water geographical landmarks: Canal, Beach, Lake, Polder
Other landmarks: Botanical_Garden, Zoo, Statue
Cultural buildings: Public_University, Theater, Public_School, Private_School, Opera, Library, Business_School, Music_School
Population: 399457 Elevation: 1.67 Continent: North America Climate: Tropical monsoon

Figure 65: Results after filtering for example 4b.

We can conclude that the filtering attributes can be useful to the user to focus his search on some specific areas when the list of alternatives presented is too large.

7 Conclusions and future work

7.1 Conclusions

The world of data mining and clustering is still a field that can be largely improved through new concepts of working with data and new algorithms to cluster the exponentially growing amounts of data, usually unclassified, that the digital era is creating.

In the DAMASK project proposed the use of semantic domain knowledge, represented in the form of ontologies, to define new methods for extracting and integrating information from heterogeneous Web resources with varying degrees of structure, performing an automatic classification, and making a semantic interpretation of the results.

Previous work in the project (Vicient, 2010) developed a new algorithm to extract information from the web using an ontology. The method created was used in the DAMASK project to retrieve semantic information on a set of touristic cities.

The final goal of the DAMASK project is the classification into some clusters the set of cities mentioned previously. This work is the explanation of the methodology used in order to classify the set of cities into a determined number of clusters, using the semantic, numerical and categorical data attributes of the cities, following some ideas previously developed in the project (Batet, 2010).

The variety of data types of the different attributes of each city presented a new challenge, since it is hard to find related investigations in the literature. The use of the K-means algorithm to cluster cities with attributes of different types is not trivial. Some previous works (Huang, 1998; Chan et al., 2004; Song et al., 2007) presented methodologies to cluster objects with attributes of different types, but there were always for uni-valued attributes. The DAMASK project comprehends objects with multi-valued semantic attributes. That is something rather new and the main problem to cluster the cities.

This work presented an algorithm to cluster the set of touristic cities considered in the DAMASK project. The results of the clustering method presented in this work can vary greatly depending on the different parameters used or the different sub-processes used in some step, like the type of OWA used when computing the semantic distance (chapter 3).

It is has to be remarked the importance of creating centroids that represent as best as possible the objects of a cluster. This process is not easy for heterogeneous data types including semantic multi-valued attributes. In this Master Thesis we have devoted great effort in studying the literature, and few and very recent approaches work with semantic concepts. Hence, find a way to **obtain general concepts that represent a set of objects using ontologies** is crucial.

A web recommender system has been developed using the aforementioned techniques. It is a system designed to be used by common user with no knowledge about the topic, making it easy to use and prepared to be used as a working prototype, which is one of the goals of the DAMASK project.

For the time being, the evaluation of the whole system is done by the research group manually. It is intended to be tested and evaluated by the tourism experts at PCTTO (Parc Científic i Tecnològic de Turisme i Oci).

It is worth to say the heterogeneity of the data of the cities makes difficult the task of finding patterns in the clustering results. For instance, one could see easily similarities in all the concepts of an attribute, but see absolutely nothing in other attributes. Hence, it can be said that the results of the clustering process are very subject to the view of each person at a certain degree. But it can be seen in the *evaluation of the results* chapter (chapter 7) that the results always present patterns that make the algorithm valid and good for production on the DAMASK project.

Of course, although the DAMASK project works with cities, the algorithm presented could work with similar kind of objects. For instance, and without leaving the field of tourism, the same process studied here could work to cluster hotels for its characteristics.

7.2 Contributions

The main contributions of this work are:

- A way to obtain a matrix of cities filled with various concepts for each one of their attributes from data extracted in previous works and using new tools to complement the information.
- A methodology to compute the degree of similarity between to objects that have several attributes of different types and several concepts for each one of the semantic attributes.
- A variation of the K-means algorithm to work with the aforementioned multi-type multi-valued objects. The results of the evaluation of the results prove their validity.
- A way to conform centroids that represent a cluster of multi-type multi-valued objects.
- A web recommender system that works from a user-profile and automatically presents a recommended cluster of objects to the users.

7.3 Future work

As it is explained previously, the system is hard to evaluate and the development of this kind of clustering is pretty new, with almost no information in the literature. Hence, the system has many things to be improved, some in the centroid creation, some in the clustering process and some others in the web recommender system.

- The use of the K-means leads to a fixed number of clusters. As seen in chapter 6, *evaluation of the results*, the resulting clusters can have sizes that vary too much. For instance, some clusters could have 40 cities and others just 3 or 4. Obviously is how K-means work, but some improvements can be done in order to convert the algorithm to a non-fixed K, separating the clusters that are too large or merging the clusters that are too small.
- Convergence of the K-means algorithm is not assured. The basic K-means algorithm for just numerical objects assures the convergence of the process. The method explained in this work can lead towards a non-ending process for some parameters like the lambda threshold and/or the initial centroids.
  - It is very important that the centroids represent as good as possible the set of objects that conform the cluster. So, it would be important to make variations on the

algorithm and make much more test and evaluation of the results to achieve the best way to obtain general concepts that represent a set of objects using ontologies.

- One of the things that are easy to recognize when evaluating the results is that almost every city (no matter the cluster that it belongs to) has some concept that is repeated for all the cities. These kinds of concepts use to have great weights, but have low discriminant values and are of low use at clustering. Is the case of, for instance, the concept football, that appears in almost every city. The use of term frequency–inverse document frequency has to be studied.
- The information extraction developed in the first steps of the DAMASK project is not perfect and there is a high number of cities that have attributes with missings (represented by the *?* symbol). The clustering and centroid creation algorithms do not work as good as it is expected when dealing with these kinds of attributes, so more work has to be done with this.

7.4 Goals accomplished

In the introduction of this work a set of goals for the DAMASK project was presented. Remember the diagram of tasks of the DAMASK project presented in the introduction:

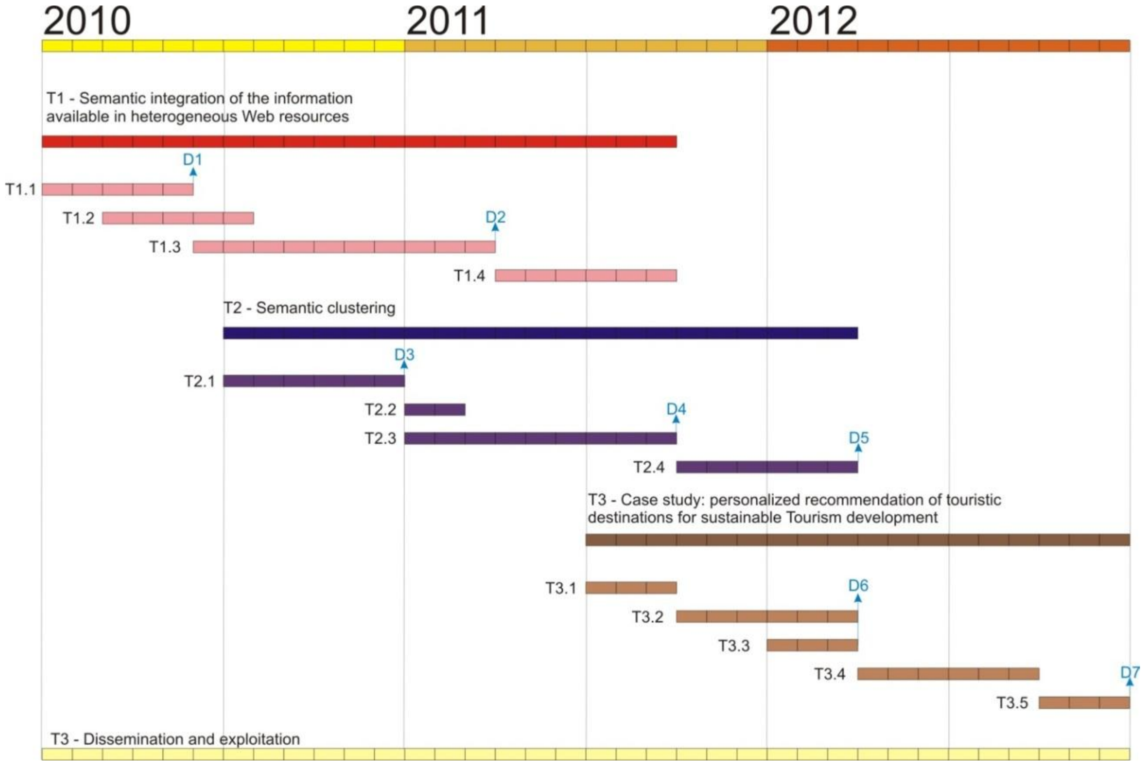


Figure 66: Tasks of DAMASK

This master thesis was done in the context of tasks T2.4, T3.2, T3.4 and T3.5 and presented the documentation corresponding as Deliverables D5, D6 and D7. Actually, chapters 2 and 3 correspond to the internal project reports T3.2 (*data matrix construction*) and T3.4 (*distance measures for heterogeneous values*), and chapter 4, 5 and 6 correspond to the Deliverables D5 (*adaptation of the K-mean*), D6 (*User-oriented recommender system*) and D7 (*evaluation of the results*) respectively.

8 References

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.

Anderberg, M.R. 1973. Cluster analysis for applications. 359, Academic Press .

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. 2010. Ontology-based semantic clustering. Ph.D Thesis. URV .

Batet, M., Sanchez, D., & Valls, A. 2011. DAMASK D3: State of the art of clustering algorithms and semantic similarity measures. Project Deliverable, URV.

Batet, M., Valls, A., & Gibert, K. 2010. A distance function to assess the similarity of words using ontologies. *Proceeding of the XV congreso español sobre tecnologías y lógica fuzzy, Huelva*, 561–566.

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.

Batet, M., Valls, A., Gibert, K., & Sanchez, D. 2010. Semantic Clustering Using Multiple Ontologies. *Proceedings of the 2010 conference on Artificial Intelligence Research and Development: Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence*, 207–216.

Beliakov, G., Pradera, A., & Calvo, T. 2007. Aggregation Functions: A Guide for Practitioners. Springer. 361.

Bremner, C. 2007. Top 150 City Destinations: London Leads the Way. *Euromonitor International*. http://blog.euromonitor.com/2007/10/top-150-city-destinations-london-leads-the-way.html

Calinski, T. & Harabasz, J. 1974. A dendrite method for cluster analysis. *Communications in Statistics Theory and Methods,* 3(1) , 1–27, Taylor & Francis .

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 .

Díaz, A. & Valls, J.F. 2000. El uso de las nuevas tecnologías en los destinos turísticos españoles. *Actas de ENTER2000*.

Díaz, P., Guevara, A., & Clavé, S. 2006. La presencia en Internet de los municipios turísticos de sol y playa: Mediterráneo y Canarias. *Actas del VI Congreso Turismo y Tecnologías de la Información y las Comunicaciones - TuriTec*.

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.

Gibert, K. & Nonell, R. 2003. Impact of Mixed Metrics on Clustering. *Lecture Notes in Computer Science*, 2905/2003, 464–471.

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.

Herrera, F., Herrera-Viedma, E., & Verdegay, J.L. 1996. Direct approach processes in group decision making using linguistic OWA operators. *Fuzzy Sets and Systems*, 79(2) , 175–190.

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 .

Lamata, M.T. & Cables, E. 2009. OWA weights determination by means of linear functions. *Mathware & Soft Computing*, 16, 107–122.

Lamata, M.T. & Pérez, E.C. 2012. Obtaining OWA operators starting from a linear order and preference quantifiers. *International Journal of Intelligent Systems*, 27(3) , 242–258.

Leacock, C. & Chodorow, M. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: a lexical database for English*, 265–283.

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.

Merigo, J. & Gil-Lafuente, A. 2009. The induced generalized OWA operator. *Information Sciences*, 179(6) , 729–741, Elsevier Inc. .

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 .

Siorpaes, K. & Bachlechner, D. 2006. OnTour : Tourism Information Retrieval based on YARS. *3rd European Semantic Web Conference*, 1–2.

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 .

Valls, A., Batet, M., & Sanchez, D. 2011. DAMASK D4: New techniques for semantic similarity measurement. Project Deliverable, URV.

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 .

Various 2008. United Nations Conference on Trade and Development. *Information economy report 2007-2008.*

Vicient, C. 2009. Extracció basada en ontologies d ’informació de destinacions turístiques a partir de la Wikipedia. Universitat Rovira i Virgili. Final Year Project, URV-ETSE.

Vicient, C. 2010. Ontology-based Information Extraction. Master Thesis, URV-ETSE.

Vicient, C. & Moreno, A. 2011. DAMASK Internal project report T3.1 Damask Ontology. Project Deliverable, URV.

Vicient, C., Sanchez, D., & Moreno, A. 2011. DAMASK D2: Ontology-Based Feature Extraction. Project Deliverable, URV.

Vicient, C., Sánchez, D., & Moreno, A. 2011. Ontology-Based Feature Extraction. *2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 3, 189–192.

Villar, A. 2007. Destinos turísticos argentinos en Internet: Un análisis de los sitios gubernamentales. *Estud. perspect. tur.*, 16(3) , 283–299.

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 .

Xu, Z. 2006. Dependent OWA operators. *Lecture Notes in Computer Science*, 3885/2006, 172–178.

Yager, R.R. 1988. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1) , 183–190.

Yager, R.R. 1996. Quantifier guided aggregation using OWA operators. *International Journal of Intelligent Systems*, 11(1) , 49–73.

Yager, R.R. 1993. Families of OWA operators. *Fuzzy Sets and Systems*, 59(2) , 125–148, Elsevier .

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. .