TIN2009-11005

DAMASK

Data-Mining Algorithms with Semantic Knowledge

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

Deliverable D7
Recommender system evaluation

Authored by

Antonio Moreno
Aïda Valls
Sergio Martínez
Carlos Vicient
Lucas Marin
Ferran Mata

Document information

project name:	DAMASK	
Project reference:	TIN2009-11005	
type of document:	Deliverable	
file name:	D7.pdf	
version:	1.0	
authored by:	A.Moreno, A. Valls, S.Martinez, C.Vicient, L.Marin, F.Mata	31/07/2013
co-authored by		
released by:		
approved by:	Co-ordinator	Antonio Moreno

Document history

version	date	reason of modification
1.0	31.July.2013	Evaluation of recommender system.

Table of Contents

1 Introduction

This document corresponds to Task 3.5 of the DAMASK project. The general goal of task T3 is to evaluate the deployment of the methods designed in the previous tasks in a particular case study: a personalized recommendation system of touristic destinations. A Web application has been designed to offer this kind of recommendations to any user. The tool is focused on searching touristic destinations in the different types of touristic resources available in Internet using the tools developed in task T1. The clustering method defined in T2.4 has been integrated in this prototype to obtain a classification of touristic destinations based on the domain knowledge and the user preferences, in order to be able to recommend the set of places that match better with the user's interests. In particular, the recommender system is built using the the data matrix explained in the DAMASK internal report 3.2, and the clusters built using the clustering technique explained in Deliverable D5. The design and implementation of the recommender system was reported in deliverable D6. In Task 3.5 an evaluation of the performance of the recommender system under different kinds of user profiles has been performed.

This deliverable is the result of the task T3.5 as shown in the schedule of the tasks in the DAMASK project is given in Figure 1. Note that this task was delayed until July 2013 due to a temporal extension of the project.

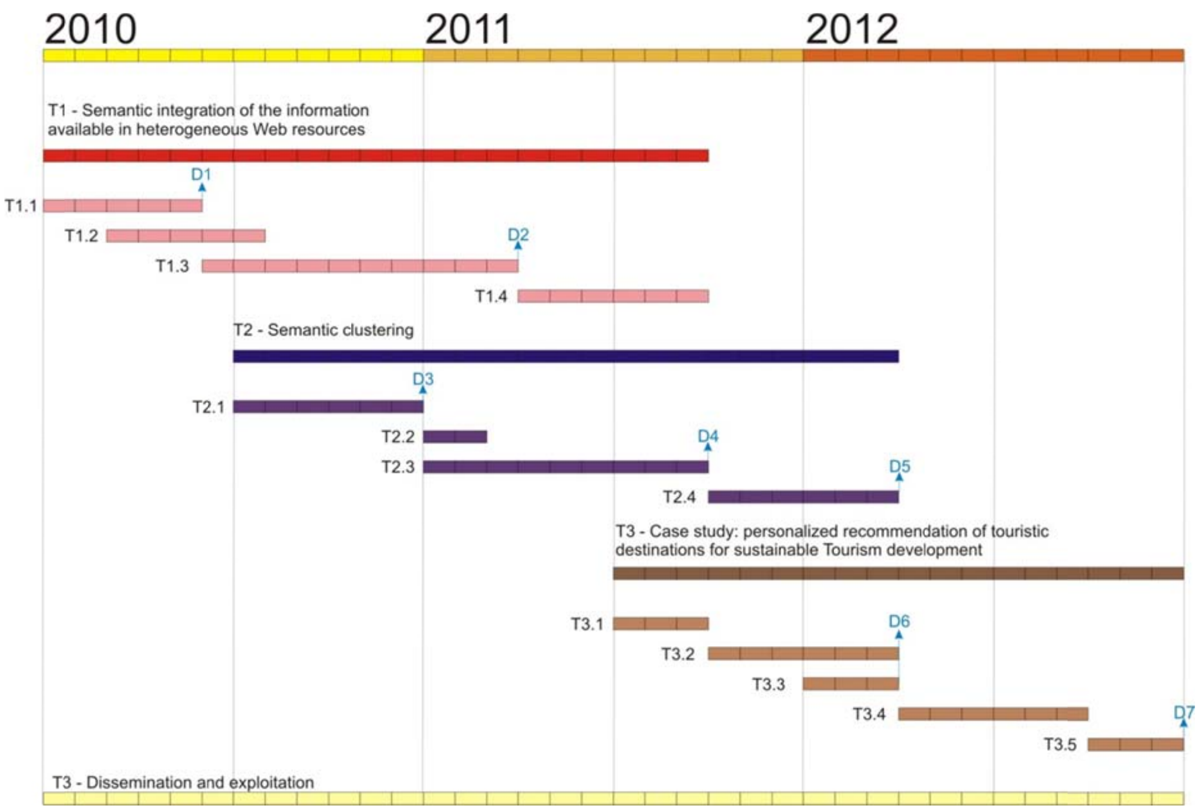

Figure 1: Tasks of DAMASK

The goal in the third task of the project was to make a personalised recommendation of touristic destinations, taking into account the preferences of the user. The cities considered by the system were grouped using the adapted k-means algorithm for which a new procedure for managing the semantic attributes was defined in the project (including the definition of a multi-valued centroid and the multi-valued semantic distance based on the Tourist ontology). The preferences stated by the user were then compared with the centroids of these clusters, and the user was recommended the cities belonging to the most similar cluster(s). The hypothesis, which was proven to be correct in the project, as reported in this document, is that the pre-clustering of the destinations allows a strong reduction on the computational cost of the recommendation process, as it is not necessary to compute the resemblance of each destination with respect to the preferences of the user, without reducing the quality and accuracy of the recommendations. The next section describes the study of the accuracy of the recommender system on different user profiles.

2 Evaluation of the recommendations

In this section the recommendations provided by our system are compared with those that would be made without the previous clustering, comparing directly the user preferences with the whole list of cities. Four user profiles have been defined to test the recommendations in different conditions (from a specific profile with very concrete requirements satisfied by a low number of cities to a very general one with wide requirements fulfilled by most of the cities). In order to numerically quantify the quality of the recommendations, we compute the F1 score that considers both the precision and the recall of the test. In this manner we are able to objectively quantify the similarity between the ideal recommendations made directly from the list of cities and the ones made by our system.

A profile is described by its preferences regarding 8 multi-valued semantic attributes related to leisure activities of touristic places (Aquatic and Nature sports, Other sports, Religious buildings, Cultural buildings, Other interesting buildings, Museums, Landmarks related to water and Geography, Other landmarks). Table 1 shows the four different profiles that have been considered in the tests and the preferences for each attribute.

Id	Aquatic nature sports	Other sports	Religious buildings	Other buildings	Museum	Water or Geographical landmark	Other landmark	Cultural building
1	--	--	--	Golf course Fort	--	--	--	--
2	Swimming	Martial art	--	Kiosk Headquarter	Military museum	--	--	Music school
3	Cycling	Ice hockey	--	Golf course	--	--	Botanical garden	Music school
4	Sailing	Football	Church	House	Maritime museum	Square	Park	University

Table 1: User profile preferences

Analysing the table of profile preferences, we can see that user 1 has selected only two values in a single attribute (he is interested in cities with forts and golf courses); thus, it is a user with very concrete requirements that only a small number of cities satisfy. User 2 provides 1 value of interest in 4 attributes and 2 values in the "*Other Buildings*" attribute. Profile 3 also has a value of interest in 5 of the 8 attributes. The fourth profile is the most general one. Not only it provides interests on all the attributes, but they are very generic and easy to find in most of the cities (e.g. *Football*, *Church*, *Square*, *Park*, *University*).

The validation of the proposal has been made comparing the results obtained by our system against an *ideal* recommendation. We assume that this recommendation can be calculated by sorting, in ascending order, the whole list of cities in function of its distance to the profile. We consider this ordered list as the ranking of cities according to this profile. The closest city to the profile (the first city in the ranking) should be the first recommendation. In order to test the behaviour of our system, we have considered a number of scenarios with different numbers of cities to be recommended (5, 10, 15 20, 25 and 30). The distance function used to order the cities with respect to a profile quantifies the distance between cities using multi-valued semantic attributes (see Deliverable 4). Notice that, since a profile is defined with the same attributes as a city, we consider a profile as a city to use with this distance function.

In the recommender system, a pre-processing step applies the adapted k-means algorithm (see Deliverable 5 for details of this adaptation) and classifies 150 cities in 10 classes of different sizes. After that, the user profile is compared with the 10 centroids to find out which of them are the most similar ones to the preferences of the user. The final recommendation is made by selecting, from the best (1, 2 or 3) cluster(s), the closest cities to the profile. The final number of recommended cities is determined by their distance to the profile. We evaluated different relative distances to determine if a city should be recommended (from 0.1 to 0.5). These distances are normalised with respect to the maximum distance between the profile and the cities of the clusters selected (which can be 1, 2 or 3 clusters). A city is recommended if its normalised distance to the profile is lower than the specified threshold, which means that it is similar enough to the user's profile.

The accuracy of the recommendations can be computed by comparing them with the ideal recommendation previously described. To do so, as in [1], we have computed the *precision*, the *recall* and the *F1* scores of the recommender system. The ideal recommendations would have a perfect precision and recall, but they would require a lengthy comparison between the user preferences and the characteristics of each of the 150 cities. The *precision* of the recommender is the ratio between the number of correct recommendations (those that appear in the *n* first positions of the list, if the aim is receiving *n* recommendations) and the number of recommended cities. The *recall* of the system is the number of correctly recommended cities divided by the total number of cities which the system should have recommended. *F1* is the harmonic mean of precision and recall.

As an example, tables 2, 3 and 4 show the number of recommendations made by our system to user 4. For each ranking of recommendations, the tables show the corresponding precision, recall and F1 values for different intra-cluster relative distances. The F1 values over 70% are highlighted in the tables.

Clusters:1	Dist:	0,1	#tcr	5	Dist:	0,2	#tcr	12	Dist:	0,3	#tcr	19	Dist:	0,4	#tcr	22	Dist:	0,5	#tcr	22
#tcir	Rec.	P	R	F1	Rec.	P	R	F1	Rec.	P	R	F1	Rec.	P	R	F1	Rec.	P	R	F1
5	1	0,20	0,20	0,20	1	0,08	0,20	0,12	1	0,05	0,20	0,08	1	0,05	0,20	0,07	1	0,05	0,20	0,07
10	2	0,40	0,20	0,27	2	0,17	0,20	0,18	2	0,11	0,20	0,14	2	0,09	0,20	0,13	2	0,09	0,20	0,13
15	4	0,80	0,27	0,40	4	0,33	0,27	0,30	4	0,21	0,27	0,24	4	0,18	0,27	0,22	4	0,18	0,27	0,22
20	5	1,00	0,25	0,40	5	0,42	0,25	0,31	5	0,26	0,25	0,26	5	0,23	0,25	0,24	5	0,23	0,25	0,24
25	5	1,00	0,20	0,33	5	0,42	0,20	0,27	5	0,26	0,20	0,23	5	0,23	0,20	0,21	5	0,23	0,20	0,21
30	5	1,00	0,17	0,29	7	0,58	0,23	0,33	7	0,37	0,23	0,29	7	0,32	0,23	0,27	7	0,32	0,23	0,27

Table 1. Number of recommended cities (Rec.) made by our system to user 4. The table also shows the precision (P), recall (R), F1 and total number of cities recommended (#tcr) compared with the total number of cities in the ideal recommendation (#tcir) for different distance values used (Dist.). In this test, only 1 cluster has been taken into account for the recommendation.

Clusters:2	Dist:	0,1	#tcr	8	Dist:	0,2	#tcr	18	Dist:	0,3	#tcr	33	Dist:	0,4	#tcr	51	Dist:	0,5	#tcr	62
#tcir	Rec.	P	R	F1	Rec.	P	R	F1	Rec.	P	R	F1	Rec.	P	R	F1	Rec.	P	R	F1
5	4	0,50	0,80	0,62	4	0,22	0,80	0,35	4	0,12	0,80	0,21	4	0,08	0,80	0,14	4	0,06	0,80	0,12
10	8	1,00	0,80	**0,89**	8	0,44	0,80	0,57	8	0,24	0,80	0,37	8	0,16	0,80	0,26	8	0,13	0,80	0,22
15	8	1,00	0,53	**0,70**	12	0,67	0,80	**0,73**	12	0,36	0,80	0,50	12	0,24	0,80	0,36	12	0,19	0,80	0,31
20	8	1,00	0,40	0,57	16	0,89	0,80	**0,84**	16	0,48	0,80	0,60	16	0,31	0,80	0,45	16	0,26	0,80	0,39
25	8	1,00	0,32	0,48	18	1,00	0,72	**0,84**	18	0,55	0,72	0,62	18	0,35	0,72	0,47	18	0,29	0,72	0,41
30	8	1,00	0,27	0,42	18	1,00	0,60	**0,75**	21	0,64	0,70	0,67	21	0,41	0,70	0,52	21	0,34	0,70	0,46

Table 2. Number of recommended cities (Rec.) made by our system to user 4. The table also shows the precision (P), recall (R), F1 and total number of cities recommended (#tcr) compared with the total number of cities in the ideal recommendation (#tcir) for different distance values used (Dist.). In this test, 2 clusters have been taken into account for the recommendation.

Clusters:3	Dist:	0,1	#tcr	9	Dist:	0,2	#tcr	20	Dist:	0,3	#tcr	36	Dist:	0,4	#tcr	54	Dist:	0,5	#tcr	66
#tcir	Rec.	P	R	F1	Rec.	P	R	F1	Rec.	P	R	F1	Rec.	P	R	F1	Rec.	P	R	F1
5	4	0,44	0,80	0,57	4	0,20	0,80	0,32	4	0,11	0,80	0,20	4	0,07	0,80	0,14	4	0,06	0,80	0,11
10	9	1,00	0,90	**0,95**	9	0,45	0,90	0,60	9	0,25	0,90	0,39	9	0,17	0,90	0,28	9	0,14	0,90	0,24
15	9	1,00	0,60	**0,75**	13	0,65	0,87	**0,74**	13	0,36	0,87	0,51	13	0,24	0,87	0,38	13	0,20	0,87	0,32
20	9	1,00	0,45	0,62	18	0,90	0,90	**0,90**	18	0,50	0,90	0,64	18	0,33	0,90	0,49	18	0,27	0,90	0,42
25	9	1,00	0,36	0,53	20	1,00	0,80	**0,89**	21	0,58	0,84	0,69	21	0,39	0,84	0,53	21	0,32	0,84	0,46
30	9	1,00	0,30	0,46	20	1,00	0,67	**0,80**	24	0,67	0,80	**0,73**	24	0,44	0,80	0,57	24	0,36	0,80	0,50

Table 3. Number of recommended cities (Rec.) made by our system to user 4. The table also shows the precision (P), recall (R), F1 and total number of cities recommended (#tcr) compared with the total number of cities in the ideal recommendation (#tcir) for different distance values used (Dist.). In this test, 3 clusters have been taken into account for the recommendation.

The analysis of the accuracy of the recommendations made to the four users is shown in Figures 2 to 5 (note that Figure 5 visualizes the results for the last user, which are detailed in tables 2, 3 and 4). These four figures show the value of F1, which is the harmonic mean of the precision and the recall (which are obtained by comparing the recommendations of the systems with the ideal list of recommendations for each user). Each of the figures has 3 graphics, which show the results obtained considering 1, 2 or 3 clusters in the recommendation process (the more clusters are considered, the bigger is the number of recommended cities, as shown in tables 2 to 4). The x-axis of each graphic represents the maximum relative distance to the profile allowed for a city to be recommended (from 0.1 to 0.5); thus, the bigger the distance, the larger will be the number of recommended cities (see also tables 2 to 4). Finally, each graphic has 6 lines, which correspond to the results obtained for an expected number of 5, 10, 15, 20, 25 or 30 recommendations.

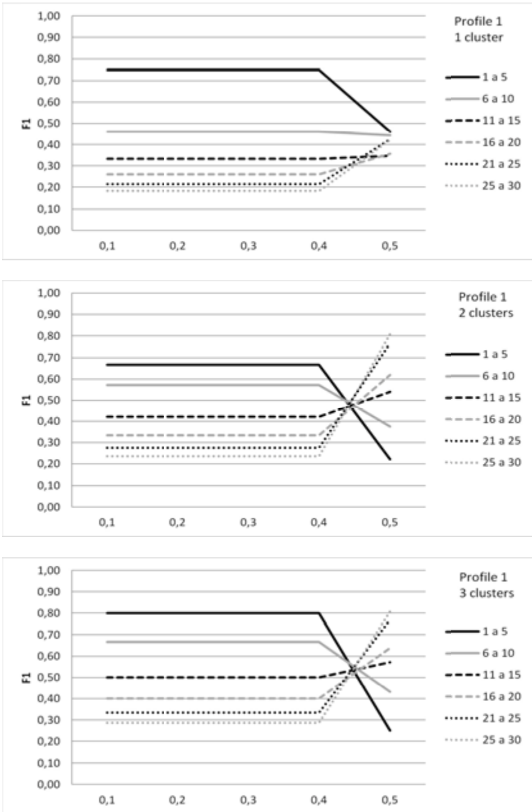

Figure 2. F1 score results of the recommendation for the profile 1

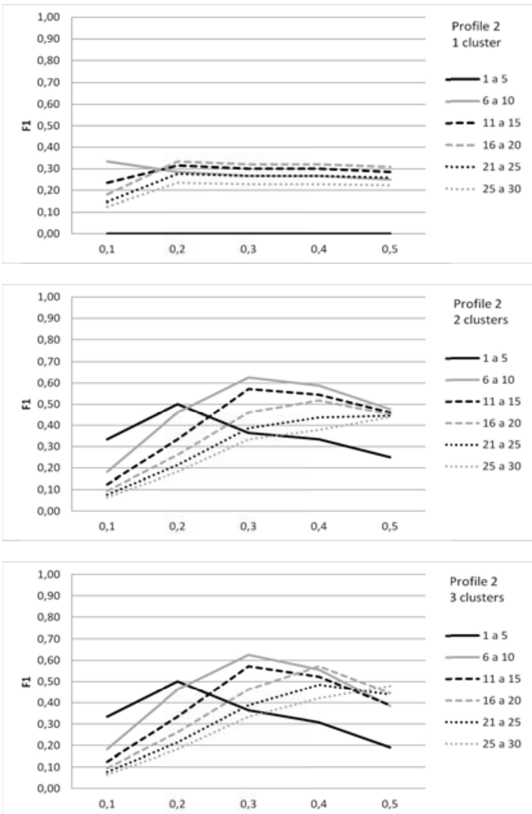

Figure 3. F1 score results of the recommendation for the profile 2

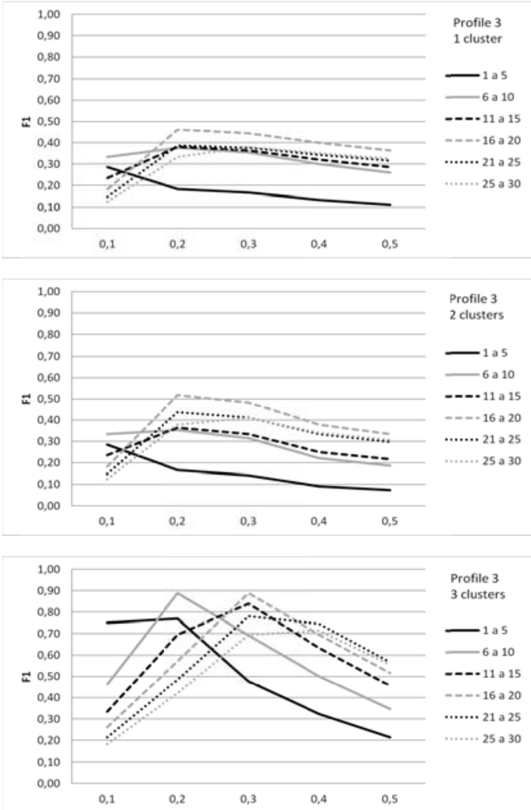

Figure 4. F1 score results of the recommendation for the profile 3

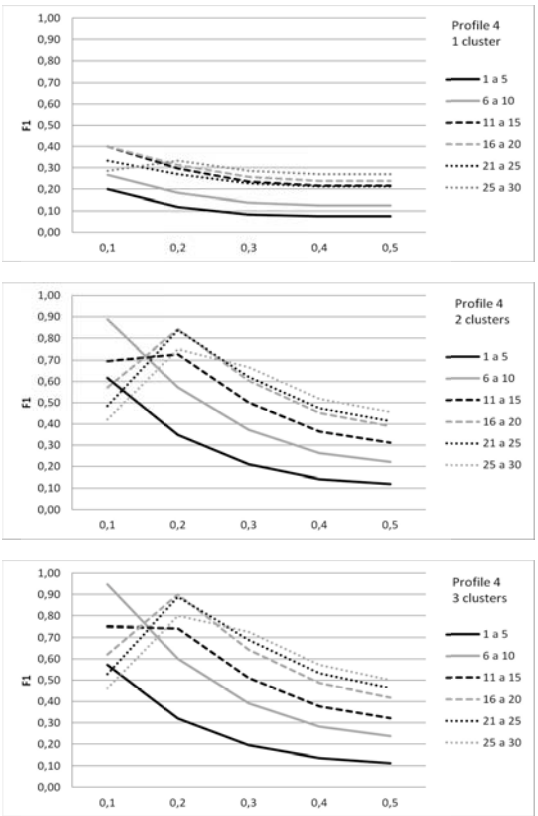

Figure 5. F1 score results of the recommendation for the profile 4

Analysing the values of F1 in figures 2 to 5, it is possible to observe different behaviours in the four profiles that have been taken as case studies. Some conclusions that can be reached from these results are the following:

- In the case of the user that presented a very specific set of requirements (profile 1), the recommender achieves high F1 values in the most restrictive setting, when it considers a single cluster, a very close intra-cluster distance of 0.1 and a low number of expected recommendations (5). Concretely, in this case the F1 score is 0.75, and the precision was 1. This fact is due to the fact that only a few cities satisfy the requirements of the user. These results show that the proposed distance measure has been able to group the similar cities in the same cluster, because taking into account only 1 cluster and the shortest distance of 0.1 the recommender has identified the cities that satisfy the required conditions. The results obtained for profile 1 also show that the F1 score is stable until a distance of 0.4 and decreases for higher values. In a similar way, the F1 score decreases when the number of expected recommendations is increased, because the system is forced to recommend more cities, including those that do not fit with the preferences of the user, reducing the precision and the recall of the recommendation.

- The results for the two users that had an intermediate set of requirements (profiles 2 and 3) are similar, but in the case of profile 3 we obtain better results for the same distance values and number of expected recommendations. The best F1 scores are obtained when the recommender uses the 3 clusters closer to the profile, maximum intra-cluster relative distances between 0.2 and 0.3 and up to 10 recommendations (concretely, a F1 score of 0.89 with a precision of 1). The explanation is twofold. On the one hand, as there are more cities that meet the requirements of the user, the precision is increased when more recommendations are made. On the other hand, as the required values are quite common and they appear in many cities (e.g. swimming or cycling), the system needs to consider several clusters and to increment the intra-cluster relative distance to find all the appropriate results. However, despite the use of more clusters and the allowance of a bigger distance, the recommender does not reduce its accuracy, which shows that the proposed distance measure discriminates correctly the cities in function of their most representative attributes.

- In the case of the user with more general requirements (profile 4), the higher F1 scores are obtained with a distance 0.1 and 10-15 expected recommendations (concretely, 0.89-0.70 for 2 clusters and 0.95-0.75 for 3 clusters, as shown in figure 5 and tables 3 and 4). There are also very good results (F1 higher than 0.70, highlighted in tables 2 to 4) with a maximum distance of 0.2 and 15 to 30 recommendations. For instance, a F1 value of 0.90 is obtained when the recommender uses 3 clusters and 20 expected recommendations (concretely, in this case the system makes exactly 20 recommendations, and 18 of those 20 cities appear in the 20 first positions of the ideal ranked list of 150 cities for user 4). However, it can be observed in Figure 5 that the accuracy of the recommendations is degraded using greater distances. The main reason is that the preferred values (park, square, house, church) are very common and they can be found in most cities. Thus, the distances between the cities are very small in this case, and it is difficult to differentiate those that fit better with the values requested in all the attributes. With an intermediate distance of 0.2 the proposed semantic measure evaluates in a suitable way the similarity between the cities.

In general, the above observations suggest that, on the one hand, in order to obtain the best recommendations with our system, the maximum relative distance to the profile allowed to recommend a city should be 0.3, since bigger distances always lead to worse results. This limit is consistent with the goal of the proposed semantic measure of evaluating the similarity between objects described with multi-valued semantic attributes. On the other hand, the appropriate number of cities to recommend is 10 and considering up to 3 clusters we obtain the best results. This fact significantly reduces the number of cities to consider initially (only 10 centroids instead of the whole set of 150 cities, 6%) and limits the number of cities to treat during the computation of the recommended cities (to the ones that belong to the 3 best clusters).

3 Bibliography

[1] C. Vicient, D. Sánchez, and A. Moreno, An automatic approach for ontology-based feature extraction from heterogeneous textual resources, *Engineering Applications of Artificial Intelligence* **26** (2013), 1092-1106.