Similar or Not Similar: This Is a Parameter Question

Andrey Araujo Masiero, Flavio Tonidandel, and Plinio Thomaz Aquino Junior

FEI University Center
Av. Humberto A. Castelo Branco, 3972
09859-901 - S. Bernardo Campo - SP - Brasil
{amasiero,flaviot,plinio.aquino}@fei.edu.br

Abstract. There is much information of users to be analyzed to develop a personalized project. To perform an analysis, it is necessary to create clusters in order to identify features to be explored by the project designer. In general, a classical clustering algorithm called K-Means is used to group users features. However, K-Means reveals some problems during the cluster process. In fact, K-Means does not guarantee to find Quality-Preserved Sets (QPS) and its randomness let the entire process unpredictable and unstable. In order to avoid these problems, a novel algorithm called Q-SIM (Quality Similarity Clustering) is presented in this paper. The Q-SIM algorithm has the objective to keep a similarity degree among all elements inside the cluster and guarantee QPS for all sets. During the tests, Q-SIM demonstrates that it is better than k-means and it is more appropriate to solve the problem for user modeling presented in this paper.

Keywords: Q-SIM, Clustering, User Modeling, Personas.

1 Introduction

The users' diversity has been growing up while the time passes [1]. The analysis of diversity in a group of users is a complex work and it usually requires unnecessary extra time of the specialist. It is important to identify a likeness between the groups to help the specialist to keep his focus on the most relevant characteristics.

It is important to determine which users are similar to each other, in order to minimize the number of profiles. To reach this understanding, Jung [9] defined a term called *Personas* to represent unique profiles. He identified that a person assumes different behaviors depending on the current scenario or on collective conventions. *Personas* was popularized by Cooper [4], defining it as hypothetical archetypes of users. This technique also could be defined as a descriptive model of users, which represents a group of real users and their features [2].

In previous research [11], a methodology to support the process to automatically create *Personas* by using clustering algorithms was proposed. The algorithm used in that case was the classical clustering method called K-Means [8]. There are several weak points on the use of K-Means for the propose to create

S. Yamamoto (Ed.): HIMI/HCII 2013, Part I, LNCS 8016, pp. 484–493, 2013.

Personas. First of all, the parameter used in K-Means is the number of groups that will be created (a k number). This parameter can turn the process imprecise due to the specialist not always knows the number of groups the data contains. In this case, the specialist needs to analyze the data and identify how many groups exist before using the algorithm, which demands an unnecessary extra time of the specialist. Another problem in K-Means is its randomness of possible results what turns the entire process unpredictable and unstable since the final clusters sets could vary according to the centroid initialization.

We look for a Quality-Preserved Sets (QPS) to define *Personas*. A QPS is a concept where a set of objects guarantees a minimum value of similarity among all objects, i.e., if the minimum value is 0.8, any element of a QPS has at least 0.8 of similarity to all others elements. QPS is important to find high quality *Personas*. If we consider *Personas* as a fictitious character that represents a group of real users [1], a group with similar elements (users) that regards a certain similarity among them is needed in order to guarantee a definition of more representative *Personas*. K-Means doesn't guarantee this similarity among all elements in a group since it depends on the pre-defined number k and the clustering process only cares about the similarity of the elements to the k centroids randomly distributed, which does not guarantee the QPS for the clusters.

In order to guarantee better representative *Personas*, this paper proposes a novel clustering algorithm called Q-SIM (Quality Similarity Clustering), which creates clusters based on the degree of similarity to preserve QPS concept. Q-SIM inverts the K-Means process and instead of require a k number of clusters, it requires a parameter Q of quality and it finds the appropriate number of clusters that guarantee the QPS for all groups based on the parameter Q indicated. The Q value is a similarity threshold that varies from 0 to 1. The Q-SIM algorithm also needs a similarity metric equation to calculate similarities between elements, like K-Means uses the Euclidean distance, for instance.

This paper is organized as following, section 2 presents the problem with k-means which motivate this research. Section 3 details the Q-SIM algorithm. Section 4 discusses the results of comparison between Q-SIM and k-means. Beyond that is presented the result of the application of Q-SIM into a project of Research and Statistic based on Digital Collection of Patient Medical Record in Center User Telemedicine (PEAP-PMPT, in portuguese) which *Personas* are created by the use of the Q-SIM algorithm. And the last two section (5 and 5) present the conclusion of work and acknowledgments that supports this research.

2 Clustering for User Modeling

During the process to create *Personas* by using K-Means clustering algorithm, a designer must decide how to overcome the problem to establish the number of cluster for input. Some works execute k-means algorithm many times with different number of cluster and compare the results. The comparison of results involves all project team and stakeholders which demand much time [1][11].

Another work involving k-means and *Personas* are presented by Weber and Jaimes [13], which use the algorithm to create an segmentation between the information search by the user. With this information, they create *Personas* to head the marketing advertisement. For defining the number of cluster, authors tried different number of cluster. The range 8 to 20 was tried and the centroids and the clusters were compared in order to determine if its necessary to merge or not some cluster to reach a low number of cluster. All these works does not guarantee to preserve the QPS for the found groups.

This problem present by k-means motivated the research of this paper and the creation of an algorithm called Q-SIM that can find the number of cluster based on the similarity value determined by the specialist and preserving QPS. The quality of a set is determined by a proximity on its elements. The proximity of elements can be calculated by a similarity metric that must be well defined.

There are many similarities' calculus. This decision depends on the project and the best way to represent database information for determining patterns into it [6]. One example is the Euclidean distance, useful for numerical data that can be represented by data points into a physical space.

Any categorical data, e.g. textual data, can be used by the euclidean distance if there is any method to convert categorical data into numeric codes. There are methods that is more related to categorical data. An example is the method presented by Dutta et al.[6].

These methods can also be combined for the calculation of similarity between objects composed by two or more variables with different types. For this kind of similarity, it is used a combination of local similarity (for each variable contained into object) and global similarity. Local similarity uses the equation 1.

$$sim(X_i, Y_i) = 1 - \left(\frac{|X_i - Y_i|}{(\max - \min)} \right) \tag{1}$$

Global similarity calculus are based on local similarity. The equation 2 define the global similarity.

$$Sim(X, Y) = \frac{\sum W_i \cdot sim(X_i, Y_i)}{\sum W_i} \tag{2}$$

Where Wi is a weight for each variable i of the object.

The result of this step is a similarity matrix between all objects. The similarity is a good calculation of the proximity of elements and it is a parameter to guarantee the QPS for the clustring process. Q-Sim uses similarity as the core parameter to find the clusters, as presented in the next section.

3 Q-SIM Algorithm

The Q-SIM (Quality Similarity Clustering) algorithm aims to automatically detect the number of suitable clusters in order to preserve the quality among all elements in a set. Q-Sim algorithm comprises 3 distinct phases: (I) Preparation of data; (II) Selection of sets; (III) Refinement of the clusters.

For the first phase, Q-SIM algorithm tries to determine groups of elements by manipulating data into sets. In this phase, Q-SIM uses the similarity matrix among objects to determine what it calls Related Sets. An object's Related Set is a set of all objects that has at least Q similarity value from the target object o. The formal definition, adapted from [12], is:

Definition 1. *(Related Set) A Related Set of object target o, denoted by $RS(o)$, is an object's set formed by following formula:*

$$RS(o \in \mathcal{O}) = \forall p \in \mathcal{O} / similarity(o, p) \leq Q$$

Where $\mathcal{O}$ is the object's set of the domain and Q is similarity value, between 0 and 1. Notice that o is part of its own Related Set since $similarity(o, o) = 1$. Although all objects into a Related Set are similar to object target o, there is no guarantee that an object p is similar to object q, when $p, q \in RS(o)$. Considering that Q-SIM looks for a QPS cluster, it must find a subset of $RS(o)$, which reaches a minimum Q value among all elements. This subset is called *Reduced Related Set*, defined as following.

Definition 2. *(Reduced Related Set) A Reduced Related Set of object target o, denoted by $RRS(o)$, is a group of objects formed by following formula:*

$$RRS(o) = \{\{c_1 \ldots c_n\} \in RS(o) / similarity(c_i, c_j) \geq Q, 1 \leq i \leq n, 1 \leq j \leq n\}$$

Notice that it is possible to exist many subsets $RRS(o) \in RS(o)$. However, the best RRS is the one that contains a biggest number of objects from original RS. Nevertheless, to choose objects from a certain group to maximize the number of objects inside RRS is hard to calculate and claim an algorithm with a high computational time. Considering that each element p of $RS(o)$ has its own $RS(p)$, any intersection between $RS(o)$ and $RS(p)$ will create a group of elements, called $RS'(o)$ that has similarity at least Q for o and for p.

As a solution to approximate the best RRS, Q-SIM uses a greedy algorithm which first locates the element $p \in RS(o)$ that maximizes the intersection $RS'(o) = RS(o) \cap RS(p)$. The process repeats recursively for the elements within $RS'(o)$, until any intersection becomes empty. The object o and chosen p values recursively obtained form the $RRS(o)$ set from the original $RS(o)$. Notice that all elements within $RRS(o)$ maintain the QPS concept based on Q value.

Q-SIM determines a RRS set for all elements of $\mathcal{O}$. Obviously, it exists a lot of intersection between all RRSs since each object has its own RRS and it belongs to another RRS from many different objects. The union of all RRSs creates the domain set of all existing objects $\mathcal{O}$.

Next phase, called Selection of Sets, is characterized to find the smallest number of subsets $RRS \in \mathcal{O}$ which comprises all objects of $\mathcal{O}$. A set of RRSs that cover all $\mathcal{O}$ objects is called $\mathcal{C}$. The problem to determine the smallest $\mathcal{C}$ is known as set-cover problem and it is proven to be NP-Complete [7]. An approximate solution used by Q-SIM is also a greedy algorithm that picks the RRS set that covers the greatest numbers of elements not covered by sets $RRS \in \mathcal{C}$. It is a good approximate algorithm for the set-cover problem and it provides a good solution near to the optimal one.

Not only the greatest number of elements is essential to choose a RRS to compose the set $\mathcal{C}$, but also a metric that show how close the elements are among them in RRS. Q-SIM uses a density function that takes into consideration the size of the RRS and the proximity of its elements. The density function definition is presented as following.

Definition 3. *(Density Function) The density of $RRS(o)$ is calculated by following formula:*

$$density(RRS(o)) = \frac{size(RRS(o))}{\frac{\sigma(RRS(o))}{\mu(RRS(o))}}$$

Where $\sigma(RRS(o))$ is the standard deviation of the object's similarities into $RRS(o)$ and $\mu(RRS(o))$ is the average of the object's similarities into $RRS(o)$. If $\sigma(RRS(o))/\mu(RRS(o)) = 0$, the density becomes $size(RRS(o))$.

With density values(definition 3) of all $RRS \in \mathcal{O}$, Q-SIM selects the RRS which has the biggest density value to be part of set $\mathcal{C}$. During this process of selection, all elements in $\mathcal{C}$ are not considered for the next selection and so on. In fact, after a selection of a RRS to compose $\mathcal{C}$, the density function is calculated again for all $RRS \in \mathcal{O}$ excluding any element already chosen and part of $\mathcal{C}$.

When a $RRS \in \mathcal{O}$ is choose by the density function, Q-SIM verifies if the objects into the RRS can be included into another existing group in $\mathcal{C}$. To do that, it is necessary to validate the Q value among all elements of both groups. If all objects in RRS can be included in a existing group, no new group is created in $\mathcal{C}$. However, if only one single object remains in the RRS, the entireRRS set becomes a new group of elements in $\mathcal{C}$ and none of its objects are inserted into another group. Since there are common elements among this new group formed by RRS and others groups in $\mathcal{C}$ and each group must be independent of others, i.e, two groups $\mathcal{A}$ and $\mathcal{B}$ are independents if $\mathcal{A} \cap \mathcal{B} = \emptyset$, Q-SIM must perform an algorithm to solve any intersections in $\mathcal{C}$ caused by the insertion of a new RRS.

A separation of groups to solve intersections problems is based on the centroids (definition 4) of involved groups. It is important the perception that each RRS is related with an object o which was responsible for the RRS formation but it is not the best representation of its group.

Definition 4. *(Centroid) Given a set of characteristics $\{c_1 \dots c_m\}$ belongs to an object o and a group of n objects denoted by $\mathcal{A}$, where $o \in \mathcal{A}$. The centroid of $\mathcal{A}$ contains a set of characteristics $\{k_1 \dots k_m\}$. It is defined as following:*

$$\forall o \ \in \mathcal{A}, \forall k \ \exists \ Centroide(\mathcal{A}), k_i = \sum_{j=1}^{n} \frac{o_j\,(k_i)}{n}, 1 \leq i \leq m$$

The process to create independent groups is presented in figure 1, where: (I) Groups with intersection are identified; (II) Their centroids are calculated; (III) Objects' similarities inside the intersection are compared with groups' centroids; (IV) The object is allocated into the group that the centroid is more similar.

The phase Selection of Sets ends when all elements of $\mathcal{O}$ are also in $\mathcal{C}$. The groups formed in $\mathcal{C}$ are the first tentative to form clusters. However, these

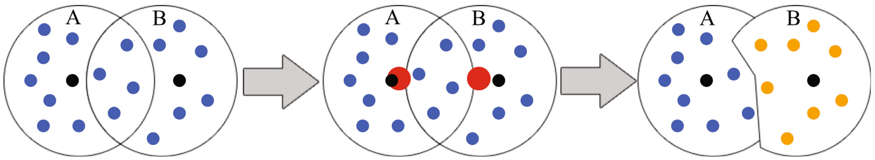

Fig. 1. Result of the process to create independent groups between the groups A and B. Red points represent the centroids calculated for each group

clusters can be optimized, what takes the Q-SIM to the 3rd and final phase: The Refinement of the clusters.

The Q-SIM performs two processes with the objective to minimize the number of groups formed and to smooth the boundaries of each one. Both processes are greedy algorithms that obtain a sub-optimal solution at most.

The first process is responsible to smooth groups' boundaries. For this process, Q-SIM compares all objects $o_i \in \mathcal{C}$ with the centroids of each existing group. If the object is more similar to another centroid than the centroid of its group, the object can be reallocated to the new group if the QPS based on Q value is not violated.

The second process is to join two or more existing groups. Q-SIM verifies if all objects of one group keep the Q value among all object of the second group. If this condition is true, these two groups are joined and becomes one.

Thus, the Q-SIM process is complete and it is possible to generate a number of groups with Q value, keeping the QPS concept. To create *Personas*, the information produced by Q-SIM is analyzed and inputted into *Personas* description. The next section presents the results obtained with Q-SIM based on data collected in PEAD-PMPT project system.

4 Results and Discussions

Before using Q-SIM to create *Personas*, it is necessary to validate its clustering processing. To validate the Q-SIM we use two database which contains data from 2-D space points, normally used in this kind of validation [10]. The results present by Q-SIM are compared with a classic algorithm of clustering k-means [8]. After analyze the results of both algorithms through the obtained graphics of clustering process, three clustering metrics are applied into results and it is discussed during this section. The three metrics used in this paper are: (I) data variance [10]; (II) Dunn's index [3]; and (III) Davies-Bouldin's index [5].

The two database used in validation process are presented in figure 2. These databases presented in figure 2, represent the most common information for clusters. The first database demonstrates a sparse data which has four solid groups. The second one is a database which is common for user information data. These 2-D databases can represent information from user of a real project.

To make the tests and compare the results betweens k-means and Q-SIM, we need to establish a pattern for analyzing the cluster result. K-means has a

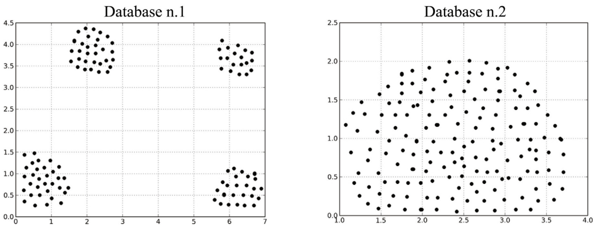

Fig. 2. Validation Databases

random process to generate the clusters, and needs to be informed how many cluster the specialist wants. The k number is a problem for some projects, mainly when the specialist wants to guarantee a degree of similarity between the user profiles and he doesn't know the exactly number of cluster into database.

For comparison reasons, we use the number of cluster found by Q-SIM,when Q value is set to 0.6, as the input k of k-means algorithm. Due to k-means randomness, we execute it 10 times and considered on the best(+) and the worst(-) results to be compared with the result of Q-SIM. The figure 3 presents the results of both algorithms for database n.1.

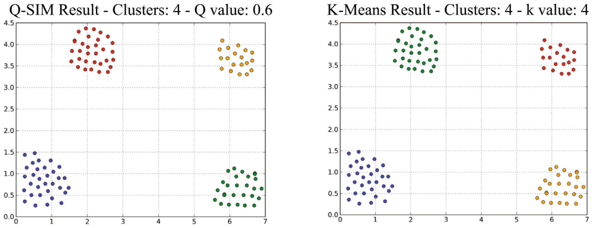

Fig. 3. Results of Q-SIM and k-means for the database n.1

Q-SIM found 4 clusters into database n.1, as expected, using the Q value 0.6 which represents 60% of similarity between cluster's elements at least. K-means also separated the database in 4 clusters, but this scenario is just possible if the specialist knows the number of clusters existing into database.

Further analysis can be made based on the indexes presented in the beginning of this section. These indexes measure the similarity between cluster's elements and how dissimilar the elements in different groups are. Figure 4 introduces the indexes result for the database n.1, for both algorithms.

Q-SIM and k-means present the same results in all indexes. The variance index measures the similarity between all elements of the cluster. The best result for variance is determined by the value closer to zero. In this case, both algorithms reach the value of 0.0014 which means very similar clusters. The other two

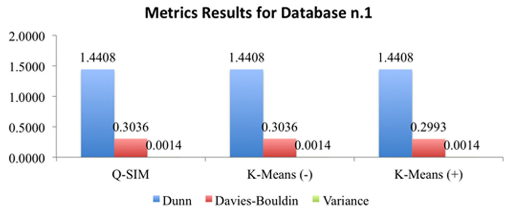

Fig. 4. Results' Metrics of Q-SIM and *k*-means for the database n.1

indexes measure how similar are the elements inside a cluster and how different they are to the others. The best Dunn's index is the one with higher value and the best Davies-Bouldin's index is the lower one. For these two indexes, Q-SIM and *k*-means reach also the same value but the best result of *k*-means reach a value 0.0043 better than Q-SIM. However, the average of both algorithms is the same for database n.1.

Continuing with the test, figure 5 presents the result for the database n.2.

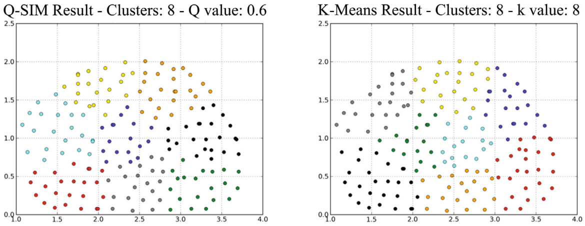

Fig. 5. Results of Q-SIM and *k*-means for the database n.2

The clusters found by *k*-means are similar to the Q-SIM clusters. However, the randomness of *k*-means can find bigger clusters than Q-SIM which may prejudice the similarity of the elements inside the clusters. To verify this situation we can analyze the indexes obtained for database n.2, present in figure 6.

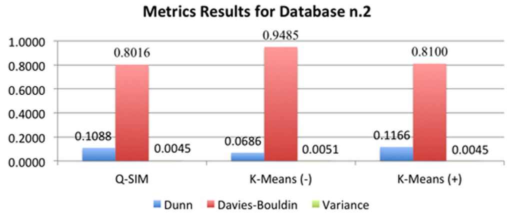

Fig. 6. Results' Metrics of Q-SIM and *k*-means for the database n.2

Analyzing the Dunn index, despite the best result of k-means are better than Q-SIM, the average result (0.092) of it is worse than Q-SIM. It shows that even with a number k of clusters defined by Q-SIM, k-means cannot always improve the quality of the result obtained by Q-SIM. For the Davies-Bouldin index, Q-SIM reaches a better result than the best result of k-means. In this index, k-means does not overcome the Q-SIM results.

The last index analyzed is the variance. The best result of k-means is quite better than Q-SIM, but k-means average result (0.0048) is a worst choice in the final analysis. After all, only Q-SIM guarantees the QPS concept in all groups.

When the information about the problems of k-means presented in section 2 is joined with the results obtained along this section, we can conclude that Q-SIM reach better results than k-means. The Q-SIM algorithm reach these result without the knowledge of specialist about how many groups exist into database and guarantees a degree of similarity among the elements of a cluster.

With these results, the Q-SIM algorithm is applied into the PEAP-PMPT project to find *Personas* based on characteristics of skills and behavior during the use of the system. The database of user information contains the following variables: (I) Time to fill text components; (II) Typing Speed; (III) Use of backspace; (IV) Number of Errors to fill a form; (V) Two or more errors in the same form; and (VI) Double click in link component.

Information of 154 users profile was collected during the experiment. We executed Q-SIM with for different Q value, 0.2, 0.4, 0.6 and 0.8. For each Q value Q-SIM find one different number of cluster, 1, 3, 3 and 5, respectively. In this case, we select the result with Q value equals 0.8 that obtained the best distribution of the user profiles into the clusters. Based on the knowledge extracted for the cluster, the *Personas* are created and help to improve the system's interface.

5 Conclusion

Based on the results, Q-SIM is the best option when compared with k-means. For the specialist to find well defined *Personas*, choosing the parameter Q is more appropriate to find groups with similar elements inside than choosing the k parameter. The similarity among elements in a group is very important in user modeling, because how much similar are the user profile that leads to the models, more closer of the real users they are. Furthermore, it is possible that the specialist vary the Q value from Q-SIM and verifies how many clusters are found during the increase and decrease of Q value. But it is important to remember that Q-SIM always will keep the similarity between elements of cluster, according with the solicitation of the specialist.

Q-SIM is not only an algorithm for user modeling, but it also can be applied into other kind of problems that need clusters for knowledge extraction. It can be used into recommend system or PICAPS [1]. Beyond these application, as future work, we want to map another variables that generate knowledge about the user and automatic improve the interface and shortcuts for the user navigation. And one more work that we are working on is the *Personas* evolution in order to identify the life time of each one and how positive is that for the projects.

Acknowledgment. To CNPq Scholarship 383677/2011-4 and PEAP-PMPT project, ref. 1465/10, In- stitutional Process number 01.10.0765.00, MCT/ FINEP/SAUDE TELESAUDE E TELEMEDICINA No - 1/2010 and FAPESP funding process number 2013/07837- 3.

References

1. Aquino Jr., P.T., Filgueiras, L.V.L.: A expressao da diversidade de usuarios no projeto de interacao com padroes e personas. In: Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems, IHC 2008, pp. 1–10. Brazilian Computer Society, Porto Alegre (2008)
2. Aquino Jr., P.T., Filgueiras, L.V.L.: User modeling with personas. In: Proceedings of the 2005 Latin American Conference on Human-computer Interaction, CLIHC 2005, pp. 277–282. ACM, New York (2005)
3. Bezdek, J., Pal, N.: Cluster validation with generalized dunn's indices. In: Proceedings of the Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, pp. 190–193 (November 1995)
4. Cooper, A.: The Inmates Are Running the Asylum. Macmillan Publishing Co. Inc., Indianapolis (1999)
5. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. 1(2), 224–227 (1979)
6. Dutta, M., Mahanta, A.K., Pujari, A.K.: Qrock: A quick version of the rock algorithm for clustering of categorical data. Pattern Recogn. Lett. 26(15), 2364 (2005)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1990)
8. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recogn. Lett. 31(8), 651–666 (2010)
9. Jung, C.: The archetypes and the collective unconscious (1991)
10. Legany, C., Juhasz, S., Babos, A.: Cluster validity measurement techniques. In: Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, AIKED 2006, pp. 388–393. World Scientific and Engineering Academy and Society, Stevens Point (2006)
11. Masiero, A.A., Leite, M.G., Filgueiras, L.V.L., Aquino Jr., P.T.: Multidirectional knowledge extraction process for creating behavioral personas. In: Proceedings of the 10th Brazilian Symposium on on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction, IHC+CLIHC 2011, pp. 91–99. Brazilian Computer Society, Porto Alegre (2011)
12. Smyth, B., McKenna, E.: Competence guided incremental footprint-based retrieval. Knowledge-Based Systems 14, 155–161 (2001)
13. Weber, I., Jaimes, A.: Who uses web search for what: and how. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM 2011, pp. 15–24. ACM, New York (2011)