Journal of Universal Computer Science, vol. 20, no. 11 (2014), 1523-1542
submitted: 11/11/13, accepted: 30/6/14, appeared: 28/10/14 © *J.UCS*

An Adaptive and Social-Aware Recommendation Algorithm for Administration Services

Luis M. Álvarez Sabucedo
(Universidade de Vigo, Spain
lsabucedo@det.uvigo.es)

Roberto Soto Barreiros
(Universidade de Vigo, Spain
rsoto@det.uvigo.es)

Juan M. Santos Gago
(Universidade de Vigo, Spain
jsgago@det.uvigo.es)

Manuel J. Fernández Iglesias
(Universidade de Vigo, Spain
manolo@uvigo.es)

Abstract: This paper addresses the recommendation of online services provided by public administrations taking into account both the specific characteristics of these services and the perception of other citizens. The solution discussed is based on an enhanced hybrid model that relies on content-based and collaborative strategies aimed to exploit the information shared by other users to validate the quality of the recommendations provided. As a relevant feature, the proposed schema takes advantage of an automatic compensation of the mentioned strategies. To make the most of theses two approaches, the use of semantics is introduced to describe knowledge and to make smart recommendation decisions. To facilitate the task of other researchers and practitioners, details about the actual development and validation of the proposed model are also included in the paper, making it possible its replication in other environmentsEurope is involved in a process of transition to digital terrestrial television that is aimed to replace all analog broadcasting infrastructures into digital ones by year 2012. Besides the substitution of all broadcasting networks scattered around Europe, this process includes the replacement of all household elements related to the reception of terrestrial television emissions, namely television appliances and antenna settings. As in any major change in the every-day life of citizens, public administrations must keep citizen informed and provide convenient support, specially when dealing with the a communication medium designated to be the carriers of services and information. This paper tackles how this situation has been faced in Galicia, a European region with special needs in this area, as shown in the paper. Through a successful use case based on Geographical Information Services and Web2.0 technologies, we illustrate some features not present in related initiatives in other areas.

Key Words: eGovernment; Semantic technologies; Decision support systems; Web-based solutions

Category: H.4, M.4, M.8, I.2.1

1 Introduction

The use of Information and Communication Technologies (ICTs) to support the development of services for the citizen has become commonplace. We have witnessed during these last years the consolidation of several so-called eTechnologies, including eLearning, eCommerce, or eHealth. However, the introduction of these applications and services in the domain of Public Administrations (here after, PAs) began later due to several circumstances inherent to the domain, like the resilience to introduce changes consequence of a tight regulatory framework or the fact that any successful solution would imply a large-scale actualization program.

We can identify the United States administration program *Access America*, launched in February 1997, as a starting point for the international development of eGovernment. Since this initial milestone, there has been a great evolution on the solutions provided for this particular domain. From initial static web sites providing information about PAs to the current integrated solutions, great amounts of effort and resources have been devoted to the development of eGovernment services and tools. Indeed, many barriers have already been overcome. Interoperability issues, accessibility problems and data models for data interchange among citizens and PAs have been adequately addressed or, at least, developers are in their way to provide stable solutions. Nevertheless, some issues are still pending, and present research activities are focused on solutions providing online services with integrated, transactional features, including advanced online platforms, and a large number of interoperable and accessible services.

This paper's scenario is related to the transition from citizens to *netizens*, that is, citizens actively taking advantage of services available on the net. Thus, this paper considers a more evolved scenario where solutions for the domain are provided in sophisticated semantic-driven environments such as social platforms. Besides, taking into account the actual dynamism of the population and the fragmentation of service providers in terms of the number of supporting administrative entities ranging from local to pan-continental or even word-wide, new needs arise. One these new issues would be the difficulty to retrieve the actual service desired among all possible options. As it has happened in the past with other domains related to the provision of digital content, eGovernment solutions should also deal with the adoption of schemes for the recommendations of content and services. To facilitate this task, the use of smart recommenders is a convenient approach. Indeed, this is the main topic of this work.

To achieve this goal, we propose a complete and detailed schema for the recommendation of services provided by PAs. The starting point for this task is to conduct a review of the current background on recommendation techniques (c.f., Section 2) and also to provide a solid representation model of the knowledge involved in our problem domain. In our approach, this representation of information will be supported by the use of semantics, as presented in Section 3. As this recommendation process is intended to be deployed as an actual software product to be tested and validated, a social network for eGovernment was deployed to serve as a validation platform, as discussed in 4. This model is presented and deeply discussed in Section 5, together with some implementation details provided in Section 6. The mechanism proposed is validated through an experimental scenario, and preliminary results are introduced in Section 7. Finally, some conclusions useful for both academic and practitioners are discussed in Section 8.

2 Background

As stated in [Ricci et al. 2010], recommender systems are software tools providing suggestions for *"items"* relevant to a given a user. They support users along several decision-making processes, such as what products to buy, what music to listen, what films to watch, etc.

Recommender systems rely on a particular recommendation strategy or algorithm that tries to estimate the utility of an item for a particular user. From a higher level perspective, the recommendation problem is reduced to the problem of estimating *"ratings"* for items that have not yet been rated by the target user [Adomavicius and Tuzhilin 2005], i.e., a recommender system implements an *utility* function:

$$utility : Users \times Items \rightarrow R \tag{1}$$

that computes a estimated level of utility of item i for user u for all $<u,i>$ pairs that have not been rated yet. Once rating for each item is computed, a recommender system is able to select the items i'_u with the highest rating, or a set of n highest-rated items, for user u, and recommend that item(s) to the user.

Different algorithms have been explored and developed in the last two decades for finding efficient utility estimators, some of which have been fairly successful, especially in the field of e-commerce [Yakut and Polat 2012]. Over the years, recommendation algorithms applied techniques from different fields such as artificial intelligence, data mining, machine learning [Kazienko et al. 2013], statistics or marketing, among many others. In the literature, the multiple and heterogeneous approaches to recommendation algorithms proposed by different authors are typically classified as:

1. *Content-based methods* [Pazzani and Billsus 2007]. In content-based methods or algorithms, the utility rating *R(u,i)* of item i for user u is estimated based on the ratings *R(u,i')* assigned by user u to items *i'* that are *similar* to item i. In a broad sense, content-based recommendation systems analyze item

descriptions (i.e. their declared features) to discover items that are analogous to those rated as relevant by the user. Thus, recommendations are made without relying on information provided by other users. These algorithms are particularly interesting in domains where there exists a comprehensive characterization of the involved items (e.g., many popular online shops use these methods), however they are difficult to use on domains where proper descriptions of items can not be obtained (e.g., videos or pictures that have not been manually described). While these algorithms have been successfully used in many contexts and domains, content-based methods suffer from two main drawbacks: i) over-specialization, when the system only recommends items that are very similar to those already rated and ii) cold-start, i.e., a new user, having few ratings, would not be able to get accurate recommendations.

2. *Collaborative methods* [Schafer et al. 2007, Sarwar et al 2001]. Collaborative or social algorithms estimate the utility rating *R(u,i)* of item *i* for user *u* according to the ratings *R(u',i)* assigned by users *u'* to item *i*, where *u'* are users *similar* to user *u*. Collaborative recommendation systems compute the similarity among different users in order to recommend unrated items to a particular user. Many approaches have been defined to measure the similarity, *sim(u,u')*, between two different users, however, due to the lack of availability of precise user profiles, the most widespread approaches have been traditionally based on the calculation of the similarity of the ratings of items previously held by both users *u* and *u'*. These algorithms do not suffer from the over-specialization situation, but they do from the cold-start one. In addition, these algorithm must face the problem of sparsity of ratings, produced when many items are rated by few users, thus hindering the characterization of the users, or when there exists users with unusual tastes.

3. *Hybrid methods* [Burke 2002]. In order to overcome the disadvantages of the two previous approaches and, above all, to incorporate their advantages in particular contexts, they may be combined in various ways into a hybrid approach to obtain intermediate performances. Combinations of methods can be produced by calculating ratings using both a content-based approach and a collaborative approach in an independent way, and combining the resulting estimations; by using some content-based techniques in a collaborative approach, and vice-versa; or defining a global approach that makes use of content-based and collaborative techniques [Christensen and Schiaffino 2014].

As is apparent from the recommendation methods that have been enumerated, recommendation systems base their operation and, hence, their performance on their ability to understand the items to recommend and/or the users to whom the recommendations will be addressed. Several techniques to improve this understanding have been applied by using existing knowledge about users and items to design a knowledge-based approach to generating a recommendation, that is, by using existing knowledge to reason about what items would better meet user requirements or desires [Burke 2000]. Particularly, along the last 10 years, several recommender systems have been proposed that use semantic technologies to characterize and easily manage information about items and/or users [Ching 2009, Wang et al. 2000]. These recommendation systems, which were initially introduced with the advent of the Semantic Web, rely on semantic modelling techniques and mechanisms such as ontologies and semantic query engines to facilitate the automatic integration and classification of information. Peis et al. [Peis et al. 2008] enumerates several semantic recommender systems focused to limit the problems of classical recommender systems by ontologically using semantic information from the categorical/taxonomical characteristics of an item or a user, or even the usage context of a particular item in a particular situation.

3 Model Description

As pointed out in the introduction, this work aims to the use of social networks [Jung and Kazienko 2012] to support the deployment of a recommender. In order to integrate collaboratively generated knowledge into the recommendation process, a solid support for knowledge representation is instrumental. For example, a comprehensive characterization of the domain is required to properly support the provision of accurate recommendations. For this, authors propose the introduction of semantic technologies, since semantics will facilitate and simplify the management, integration and query of heterogeneous information.

However, the provision of a generic model for the description of services in the context of PAs is not a simple task. This situation becomes apparent if we review models and proposals from different institutions and international projects. Indeed, we cannot find in the literature a common, agreed-upon approach, or platforms broadly accepted as reference models.

Several levels of semantic complexity are to be addressed in the framework of this proposal, ranging from fully articulating ontologies to informal user-generated folksonomies [Van der Wal 2009]. To describe the knowledge related to PA services and citizens, this work relies on the use of high level semantic models, and more specifically formal ontologies represented using OWL [W3C 2009]. Furthermore, to get the advantages of collaborative knowledge construction mechanisms, folksonomies are also introduced. These ideas are applied to the characterization of both users (i.e., citizens) and administrative services available (ASs).

In the context of this work, we introduce a new formulation of the interaction between the citizen and the PA supported by a formal semantic model

[Alvarez-Sabucedo et al. 2010]. As all interactions between them are driven by the exercise of a right or the fulfilling of an obligation, we propose the definition of services provided by the administration in these terms. In turn, this will enable us to focus on what the citizen is requesting and not on the PA offering the service. As already mentioned, we will exploit the power of OWL to express formal information. Nevertheless, we must keep in mind that OWL is just a tool to express knowledge with all its potential and limitations. As developing an ontology is a fairly common task in the field of knowledge engineering, different construction methods have been proposed. Among them, Methontology [Fernandez-Lopez et al. 1997] has been the methodological model chosen as the basis for the construction of our ontologies. This methodology proposes several stages and phases to construct an ontology in an organized manner.

Instead of developing services in a data layer directly from use cases expressed in natural terms, the provision of a semantic description of services is pursued at a higher level in a scalar and reusable manner. To model this idea, we introduce the Administrative Service (AS) concept. This concept defines any single service that a given PA is providing to citizens to fulfil a request from them. Any common service is eligible to be represented under this concept (e.g., getting a transport discount card, paying a fine or a tax, requesting enrolment on a public school, etc.).

An OWL-based model is provided to characterize all the relevant features involved in an AS (e.g., description of the task to be performed, public administration supporting the AS, required documents, documents generated upon completion of the task, service area, cost or taxes involved, etc.). Also ancillary classes are included to model geographical regions, locations and addresses, time, etc. Additionally, several properties have also been identified regarding ASs. They support the implementation of mechanisms to discover which AS may better fit to a specific user need, or to check the correctness of the knowledge included. For example, with these additional properties it is possible to assess that every AS generates some Document, or that every AS is supported by a single PA. Obviously, further details about the conformance to local or national laws regarding documentation and legal procedures are no considered at this point, and actual implementations of the system in specific administrations should take care of it.

A similar approach is considered for citizen. Using semantics as the formal support, a definition based on FOAF is derived for its characterization. This feature will play a main role later for facilitating the recommendation as process as we will take advantage of the overlapping information between the latter and services available.

This proposed model can be considered as suitable for the domain of eGovernment due to some specific circumstances of this domain [Jaschke et al. 2000], namely all operations require some input documents, the most common output in the service is a new document, there is no need or opportunity for bargaining about services, there are very explicit limits and conditions about data management in terms of trust and security (i.e., non-repudiation, privacy, integrity and confidentiality), and operations do not have real time constrains.

It may be argued that because of the very nature of services related to public administrations, service recommendations in the framework of eGovernment would have a limited impact. After all, if a citizen needs to complete some procedure to fulfil a requirement or to exercise a right, in most cases there will be just a PA involved, and probably a single and specific AS. For example, it doesn't look logical if there were several options to obtain a driving license or to pay an overdue tax. However, in most cases, no matter procedures are strictly defined, they or the associated requirements may depend on the citizen's profile (e.g. retired persons, entrepreneurs, parents of large families or disabled persons my have specific requirements or may have access to specific adapted procedures, in some cases in an optional basis), and different procedures may fulfil the user or the administration needs as side outcomes (e.g., to report a change of address may trigger a renewal of an ID card).

In the context of this work, the use of folksonomies, and more specifically the collections of tags directly provided by the citizens, will play a fundamental role. Users are expected to assign tags to services (i.e., ASs), and these tags will play a two-fold function. They will be used, as intended, to describe the service itself, but they will be also used to describe users themselves. Thus, the tags assigned by a citizen to a service will also be included in the personal profile of this user, as it is understood that these tags show the citizen's interests in an indirect way. This information is used in the recommendation scheme as discussed below.

The tags provided under this schema lack any actual structure or internal relationship. They may be considered as just a collection of labels assigned to services by citizens, and to citizens. Their expressive power relies on the words used as labels assigned directly to the resources, and on their relation with other resources under the same tag. Obviously, there is a clear and inherent dependence with the language used, which doesn't occur in the case of (formal, language-independent) OWL-based descriptions.

The knowledge to support decisions in this context comes from the number of times the same tag is bound to the same AS, from which tags are used by each citizen, and from how these tags are distributed among candidate ASs. As shown above, taking into account these parameters, it is possible to guide decisions and assist users in their decision-making processes related to public administration services.

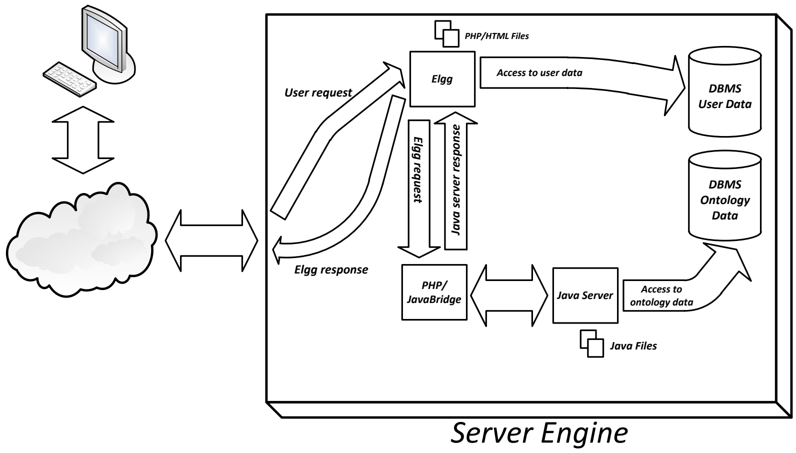

Figure 1: Sketech of the proposed model

4 Administration Services Web Portal

A Web portal for eGovernment services in line with the proposed model was deployed to properly host the environment for this recommender. The web site hosts services provided by PAs according to the discussed semantic model, and users of the site are provided with tools to properly interact with them. Additionally, these services are presented to the user taking advantage of existing social network features. Thus, citizens, once logged into the system, may access to their profile (i.e., information about themselves) and manage the tags and ASs that may be relevant to them (i.e., information about the resources available).

The tools provided were implemented as Elgg [ELGG 2008] widgets. Elgg is one of the most popular platforms to deploy custom social networks. In line with the model presented, the provided solution manages information expressed by means of an ontological model. Ontology support was provided by the Jena library [Jena 2009]. It was also necessary to provide a convenient interface to support communication between the Jena Java back-end and the Elgg PHP front-end. For this, we relied on the PHP/Javabridge [JavaBridge 2010].

Several software components are orchestrated to fulfil all the requirements of the system, as outlined in Figure 1. On the top of a semantic database (i.e., a knowledge base) where all the information is stored, a semantic engine operates on the data elements. According to the identified business model, this engine is provided as a Java-based software module that uses Jena and additional specific libraries developed for this project to implement the different system functionalities. A XML streaming-based communication channel supports the exchange of information between the Java web server and the PHP front-end.

The exchange of information between the Java web server and the PHP front-end is supported by a XML streaming-based communication channel. More specifically, a Tomcat server runs as servlet instances the Java clases invoked at the PHP modules in the front-end. The artifact supporting this communication is a J2EE web application.

Final users, that is, citizens, are intended to use the system as a gateway to locate and invoke services from PAs. In order to access the system, citizens require just a web browser. All system features are provided by the implemented Elgg widgets.

Users willing to get involved in the system are requested to provide some personal information together with some information related to their particular context such as areas of interest and geographical scope. Using the provided tools, the citizen can take advantage of the social features of the system to locate services and interact with other users. The system introduces tools to highlight ASs and also to tag them. Those ASs are considered to be part of their personal profile in the system, and are used to discover knowledge about users themselves and other users similar to them. The system also provides the actual recommender interface by means of an ad-hoc widget. This widget implements the recommendation algorithm described below.

5 Recommending Services

As already pointed out above, the recommendation approach is organized into a two-phase scheme. The aim is to take advantage of the two types of semantics used within the project, that is, light-weighted and heavy-weighted semantics. Furthermore, our recommender is personalized, as it makes personal recommendations for each citizen according to their profile and the current knowledge of the system about services and preferences. In other words, all filtering processes and ranking operations will be conducted in an individualized manner for each citizen.

In our schema, the first step involves heavy-weighted semantics, i.e., the knowledge expressed using OWL. In this stage, a pre-filtering is carried out and only ASs that could eventually be of any use for the citizen are included on a pool of services to be ranked. Citizens should provide some information beforehand about themselves. This information is enriched with information about those ASs tagged and highlighted by the user. All this information eventually becomes part of the citizen profile, and it is handled through the ontological support module as discussed above. Taking advantage of the support for actual reasoning in OWL and some ad-hoc limitations from the domain, some pre-conditions can be set, such as selecting only services available for the region where the citizen is living, or just those services that may be invoked using the documents in possession of the citizen under consideration. Differently from other domains (e.g., eCommerce), criteria for the eligibility of a particular AS are established by legal reasons and not by users' wishes. Therefore, it is possible to pre-filter services to discard ASs that could not be invoked after all.

After this initial step, it is obtained a pool of candidate ASs that could be relevant for a given citizen. The next step consists on constructing a ranked list including all relevant ASs to be presented to the citizen. Note that this raking is specific and independent to each registered user/citizen.

Service ranking is performed according to a hybrid algorithm taking advantage of both content-based filtering for the generation of recommendations taking advantage of former habits and current contents; and collaborative filtering for the provision of recommendations using similar users' previous selections. This proposal also introduces an additional criterion to overcome, to a certain extension, the limitations due to the lack of related information about the user. An additional factor is introduced to promote those ASs that are highly demanded. The idea is to take advantage of the remaining information in those scenarios where no information close to the user is available. We can take for granted that this information will actually be there as the platform has already some background, i.e., knowledge generated by pre-existing users that have already generated tags bound to ASs or AS invocations. This approach is further discussed along the next paragraphs.

During the second stage, a ranking of results is constructed by using the folksonomies generated by users as a consequence of their interactions with the system. We propose a hybrid model based on a combination of content-based filtering, collaborative filtering, and information about the current status of the system.

The formal representation of this idea is collected in the formula below:

$$\begin{aligned} ASRank(cit_i) = {} & \alpha_i * (S_{1,i} + S_{2,i} + S_{3,i}) + (1 - \alpha_i) \\ & * \sum_{\forall cit_k \sim cit_i} (S_{4,j,cit_k} + S_{5,j,cit_k}) + S_6 \end{aligned} \tag{2}$$

Note that this formula is evaluated for each AS under consideration (i.e., each AS that has passed the first filtering stage) and for each citizen. Parameters depending on a particular citizen include the reference to that citizen by means of the subindex j.

In Equation 2 a number of coefficients are included. These coefficients are grouped according to the following criteria:

Content-based filtering. This is considered in terms S_1, S_2 and S_3. These terms are affected by parameter α_i. This component involves recommendations from former selections of the current citizen. This way, the mentioned components respond to the following concepts:

- Recommendations based on the citizen cit_i and its tags in S_1 regarding AS_k:

$$S_{1,i} = sim(AS_k, cit_i) \tag{3}$$

- Recommendations based on previous ASs tagged by the citizen in S_2:

$$S_{2,i} = \frac{\sum_{\forall AS_p \in AS_{\text{taggedByCit}_i}} sim(AS_k, AS_p)}{N_{\text{taggedByCit}_i}} \tag{4}$$

- Recommendations based on previous ASs selected as preferred by the citizen in S_3:

$$S_{3,i} = 10 * \frac{\sum_{\forall AS_p \in AS_{\text{highlightedByCit}_i}} sim(AS_k, AS_p)}{N_{\text{highlightedByCit}_i}} \tag{5}$$

Collaborative filtering. In particular S_4 and S_5, that is, terms depending on $(1 - \alpha_i)$. The similarity between two citizens is measured as the Euclidean distance between the vectors containing the tags used for their description. Due to practical considerations, factors $S_5 = 10$ and 0.75 as the experimental threshold to consider two citizens as similar were established.

- S_4 for those ASs tagged by a similar citizen:

$$S_{4,i,cit_k} = f_{i,k} * \frac{\sum_{\forall AS_k \in AS_{\text{taggedByCit}_k}} sim(AS_j, AS_k)}{N_{\text{taggedByCit}_k}} \tag{6}$$

- S_5 for those ASs marked as highlighted by a similar citizen:

$$S_{5,i,cit_k} = 10 * f_{i,k} * \frac{\sum_{\forall AS_k \in AS_{\text{highlightedByCit}_k}} sim(AS_i, AS_k)}{N_{\text{highlightedByCit}_k}} \tag{7}$$

Both components, $S4$ and $S5$ include a weighting factor $f_{i,k}$ that is defined as the similarity between citizen i and citizen k.

Recommendations from the system. The proposed model includes an additional factor based on what could be considered as *hot* ASs. This factor helps to overcome some of the drawbacks from both content-based and collaborative filtering. In particular, factor S_6 is modeled as follows:

$$S_6 = \frac{\text{Invocations for } AS_k}{\text{Max invocations for any AS}} + \tag{8}$$
$$\frac{\text{Highlighted times for } AS_k}{\text{Max times highlighted any } AS} + \frac{\text{Tagged times for } AS_k}{\text{Max times tagged any AS}}$$

All these S_i parameters are normalized to fit in the range $0 \dots 1$ in order to ensure the equilibrium on the factors included. The similarity factor is used in a consistent manner along the proposal, i.e., the approach used in this frame is always the initially proposed Euclidean distance. Also, note that content-based and collaborative parameters are respectively modified by α_i and $(1-\alpha_i)$. These parameters are introduced to balance the relevance of the content-based and the collaborative contributions to the final result and it is a innovate idea in this context. From the review of the literature, it can be seen that both approaches has their own drawbacks. The idea is to provide some sort of balance to make the most of the available data. Thus, the problem seems to be how to estimate this value to maximize the utility of the results.

In our approach, the computation of the α_i is done in a fully automatic way. To find the optimal value for this parameter, additional information from the user is brought into scene, namely the highlighted ASs. Using this information, we will consider that the algorithm is performing the best, the more highlighted ASs are among the top system recommendations. Therefore, a discrete set of values for α_i is explored and the value with a greater number of highlighted ASs among the recommendations is selected. Therefore, each $\alpha_i$has a variable value for each citizen depending on the amount of information provided by them, and the information from related citizens in the system. These calculations are validated by means of an experimental scenario using volunteers to test the system.

There is an extra citizen-dependent factor S_6 that is not affected by the α_i. It is introduced as a startup seed whose value will only be relevant when both content-based and collaborative values are low, that is, when there is quite little information about the user in the system. Thus, when a new citizen is registered, this parameter is capable of supporting initial recommendations based on the overall behavior of the system.

6 Implementation Details

In order to implement the formula above, several parameters and coefficients must be estimated. As the reader can note, cross dependencies exist among them. Performing all required computations involves a large computational cost, as a complete set of value has to be computed for each citizen. The computational requirements rapidly increase with the number of citizens due to the existence of both content-based and collaborative components that require crosschecks among all citizens. To address this issue, implementation is split into two different processes, namely a batch process running in the background, and actual ranking computation.

The batch process is run during low occupancy periods, around midnight in our case, to harvest the different pieces of data required to run the complete algorithm. As shown on Figure 2, this process fetches most of the required data for the final ranking. Note from the definition of $ASRank$ above (cf. Eq. 2) that some factors in the recommender need a large amount of data from a large amount of users. Even if data pre-filtering has been already performed (i.e., only citizens and ASs fitting some semantic conditions will be taken into account), all these calculations would demand too much time to be performed in interactive mode. To guarantee an adequate response time, this process is scheduled to recover this information once a day to make it available when required for actual ranking estimation. However, the batch process does not recover all data, but only information that can be considered as being stable, that is, information whose daily changes will have a limited impact in terms of the final calculations. This would not be the case, for example, for the tags used by a citizen, as this information may dramatically change within a session, which in turn may have a high impact on the ranking outcomes.

To sum up, when ranking generation is requested by a citizen interacting with the Web platform, the algorithm in 3 is launched. This algorithm takes as inputs the outputs of the batch process outlined above together with other required data. The content-based part and the collaboration contributions to the final result are separately computed. Then, the best α_i is estimated, and the final ranked list for the citizen is computed.

The estimation of the best α_i for each citizen is conducted using a specific procedure (cf. Fig. 4). This algorithm takes as inputs the values of the content-based and collaboration components, the value already estimated for S_6, and a certain value n defined as the number of possible different values for α_i. The idea behind this strategy is quite straightforward: for each of the n values, the final ranking is calculated. For each of them, the number of already highlighted ASs is taken as an estimator of the quality of the recommendation. The empirical reason for this relies on the idea that the services considered as relevant by citizens themselves should always be included in the ranked list. Therefore, the more ASs are included in the system's top 20, the better the recommender's performance is considered. With the obtained best α_i value, the ranked list is eventually returned to the citizen. Actually, this new rank is not re-computed, but retrieved from the α_i computation phase. Additionally, the highlighted ASs are removed from the final recommendation, as it will be redundant to suggest new services that are both already known and highlighted as relevant by the citizen.

7 Validation

The validation of this sort of tools is complex as it is not possible to provide a formal prove of the effectiveness of the final result. Therefore, to assess the

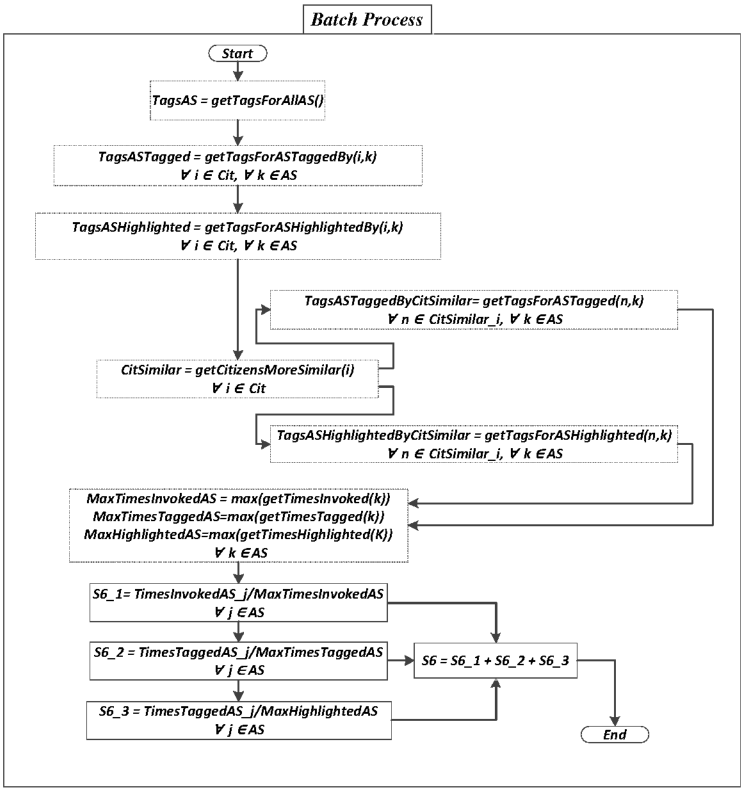

Figure 2: Background process for numerical optimization

behavior of the proposed ranking algorithm we opted for an empirical validation of the quality of the results based on the perception of final users. We selected a group of 50 users to interact with the system according to the model described in this paper. Users were asked to look for ASs to use them and tag them according to their perception. Up to 200 ASs were manually introduced in the system from actual services provided by municipalities, the regional government of Galicia and the Spanish government. The selection of both PAs and ASs was done according to the origin and characteristics of test users to ensure the appropriateness of the test-bed.

After four weeks of free usage, we selected among the test users who have interacted with at least ten different ASs, the six ones who had created more tags. This group of six users was asked to validate the model.

Each user was given a description of five situations involving the interaction

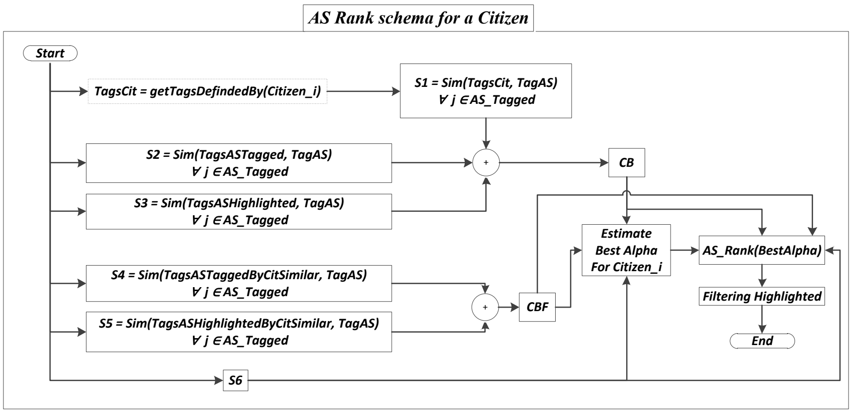

Figure 3: Algorithm for estimating the rank of ASs for each citizen

with a PA to gain access to some ASs (the same five situations were proposed to all six users), and they were asked to use the search facility in the application to obtain a ranked list of services using the proposed algorithm. Then, they were asked to analyze and access the services proposed by the algorithm, and to construct their own ranked list according to their own perception. In other words, each of the six users performed five queries to obtain five ranked lists, and constructed five additional ranked lists according to their own perception.

Thus, for each of the queries performed we obtained a pair $(l_a,\ l_p)$, where $l_a = (as_1,\ \ldots\ as_n)$ is a ranked list of ASs returned by the algorithm, and $l_p = (as_j)$ where $j = i(1) \ldots i(n)$, $i(k) \in \{1, \ldots, n\}$, $i(k) \neq i(l)$ for $k \neq l$, is a ranked list with the same elements in l_a, only ordered in a different way according to the perception of the corresponding user.

For each pair above, we defined a pair of vectors in N^n as follows:

$$v_a = (1, 2, \ldots n)$$
$$v_p = (i(1), \ldots, i(n))$$

Note that v_p is a vector in N^n whose i-th component is the position in the original (i.e., algorithm's) ranked list of the AS that the user perceived to be in position i. For example, $v_p = (1, 2, \ldots, n)$ means that the user perceived that the ranked list returned by the algorithm already had the correct ordering, and $v_p = (3, 2, 1, \ldots, n)$ means the the user perceives that the most relevant AS

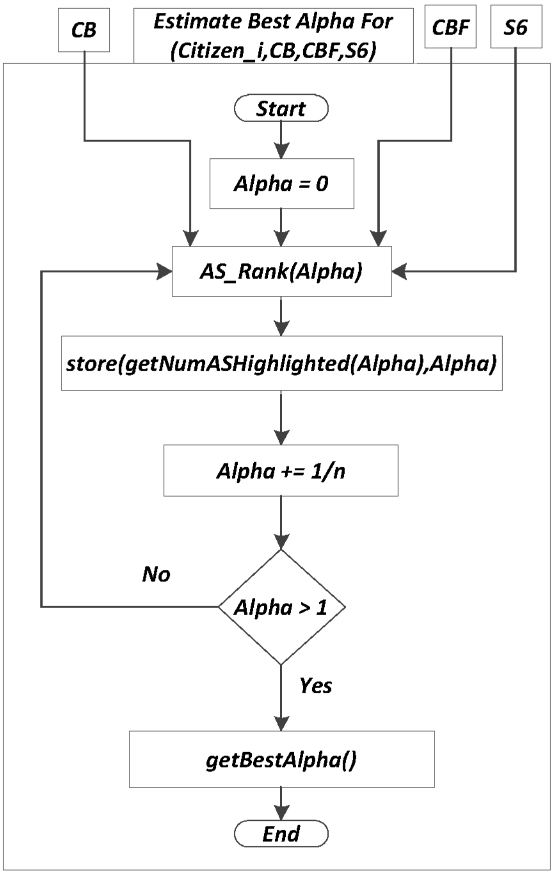

Figure 4: Algorithm for finding the best α

would be the third returned by the algorithm, and that the first returned would only be the third most relevant.

We computed the euclidean distance between the elements in each pair of vectors above. Note that this works as an indication of the *similarity* or the *degree of matching* between the ranked list returned by the algorithm and user perception. The distance will be 0 if the user perceived that the ranking provided by the algorithm is correct, and will grow both with the number of ASs perceived as incorrectly ranked, and with how far a given AS has been placed from its perceived correct position.

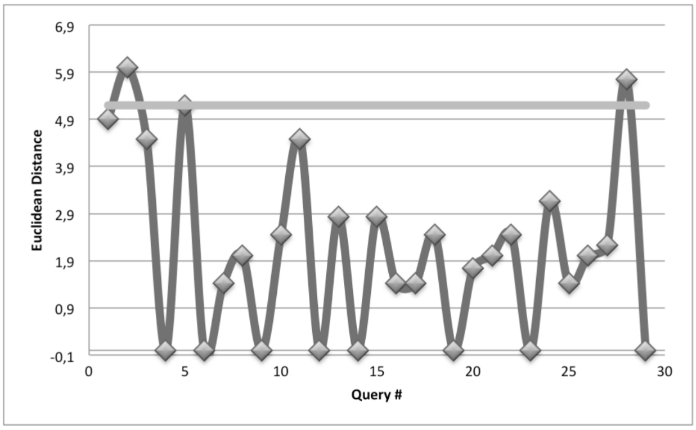

Figure 5: Algorithm vs. users' perception for best 3 results

We computed this distance for $n = 3$ and $n = 5$, that is, we took into account only the first three (resp. five) ASs returned by the ranking algorithm, and the corresponding user-perceived ranking. The results are collected in the graphs in Figs. 5 and 6 respectively. As a reference, we included in the graphs the distance corresponding to the following perceived rankings:

– For $n = 3$, $v_p = (4, 5, 6, 1, 2, 3, 7, \ldots)$, therefore $d_{r,3} = 5.1962$

– For $n = 5$, $v_p = (6, 7, 8, 9, 10, 1, 2, \ldots)$, therefore $d_{r,5} = 11.180$

These distances are represented with a clear horizontal line in each of the graphs.

For $n = 5$, the distances measured show that, in most cases, the best ranking as perceived by the users is quite similar to the ranking computed using the algorithm. The values in Fig. 6 correspond to perceived rankings with only minor changes in few ASs, to be relocated close to their positions in the original list. Results for $n = 3$ are also fairly positive. Note that to pass the reference threshold $d_{r,3}$ it is sufficient that just the three first ASs in the original ranking to be incorrectly placed. In most cases, at least one of the first three ASs is on or very close to the original position as perceived by the user.

8 Conclusion

The eGovernment domain is currently undertaking a path towards more mature solutions. As it has already happened in other environments, tools will evolve to achieve a higher level of sophistication and usefulness. At the present time, the

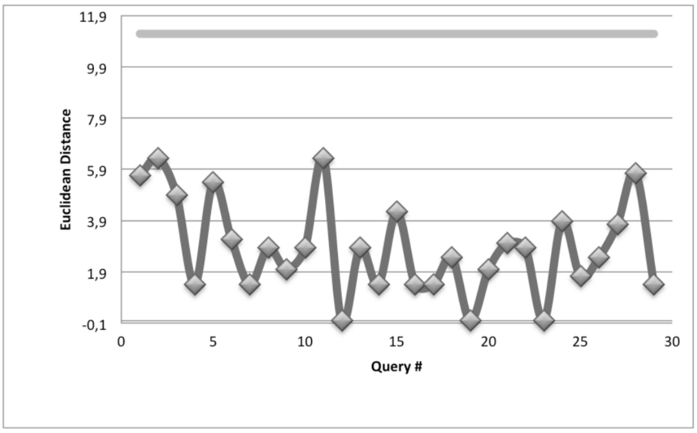

Figure 6: Algorithm vs. users' perception for best 5 results

eGovernment domain is tackling situations already solved in other environments. In this line, the provision of support for recommenders may not innovative in other domains. However, to the best knowledge of the authors, there are quite little contributions regarding this issue in the domain of eGovernment.

Due the proliferation of services from PAs, the expected increasing level of service interoperability, and the increasing participation of citizens in online administration procedures and services, it is becoming more and more relevant the provision of some support to locate in a simple way the most appropriate services to a given citizen's profile. This work is intended to contribute to fill this gap. In our case, we took advantage of existing results in the field of smart recommendations to adapt them to the eGovernment domain to provide integral support to the recommendation of administration services to the citizen.

The problem to recommend a single service or a set of services among the entire pool of services is not new. Many domains have already addressed this issue and some relevant applications support the recommendation of music, movies, books, etc. Nevertheless, in the domain of eGovernment, the nature of recommendations is different. For instance, some administrations exhibit an obvious lack of competence for fulfilling some services, not all services are possible for legal constrains, etc. Thus, the task of recommending services is better defined in this domain as the task of discovering services that may potentially be of the interest to the citizen, taking into account all the different administration bodies (local, regional, national, sectorial) that may be legally competent to address each specific situation.

This paper proposes a two-phase algorithm aimed to improve existing solutions for the recommendation of services in the eGovernment domain. An initial phase based on heavy-weight semantics is conducted to filter out those services that are not likely to be used, that is, those that according to the OWL-represented knowledge are not applicable because they belong to a different area, correspond to non-related public administrations, or violate other requirements addressed in OWL. In a second phase, the remaining ASs are ranked according to a hybrid scheme designed to promote those services that are likely to be interesting to each specific user according to the (user-defined) tags involved. The main lesson learnt from this approach is the synergies that can be generated from the cooperation of heavy- and light-weight semantic annotations. These technologies are not substitutes of each other, but complementary approaches to tackle complex semantic-driven situations.

According to the preliminary experimental results, the outcome is a fairly accurate mechanism for the recommendation of services within the context of public administrations. The ranking algorithm is quite balanced, as it overcomes the main limitations of other models described in the literature [Wozniak 2011] and does not require an extended background on the user to make accurate suggestions.

From the point of view of software development and implementation, several lessons could also be learnt from this contribution. Presently, the use of a Java-based framework for handling semantic data is preponderant and, therefore, its integration with other online tools and environments should be considered. In our case, the use of an open off-the-shelf solution to link Java-based and PHP-based code turned out to be satisfactory approach. Nevertheless, some issues regarding efficiency and security must be addressed in order to implement and eventual real-world recommendation service based on this approach.

The authors are currently working in the application of bayesian neural networks to estimate the best value for the α_i. As there is a certain set of inputs and fixed criteria to evaluate the output, it seems quite likely that this approach could play a relevant role in this scenario. Nevertheless, at the time of writing this document this option has been only barely explored.

In the opinion of the authors, public administrations must also consider their future portfolio of services under the perspective of an interoperable framework. Thus, services should be defined under the support of a semantic interoperable layer where features such as service federation, composition and orchestration can be supported in a transparent way for the citizens.

References

[Adomavicius and Tuzhilin 2005] Adomavicius, G., Tuzhilin, A.: "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions"; IEEE Transactions on Knowledge and Data Engineering 17(6), 734-749.

[Alvarez-Sabucedo et al. 2010] Álvarez Sabucedo, L. M., Anido Rifón, L. E., Corradini, F., Polzonetti, A., Re, B.: "Knwedge-based platform for egovernment agents: A web-based solution using semantic technologies"; Expert Systems with Applications 37(5), 3647 – 3656.

[Burke 2000] Burke, R.: "Knowledge-based recommender systems"; Encyclopaedia of Library and Information Systems 69(32).

[Burke 2002] Burke, R.: "Hybrid recommender systems: Survey and experiments"; User Modeling and User-Adapted Interaction 12, 331–370.

[Ching 2009] Hsu, I.-C.: "Sxrs: An xlink-based recommender system using semantic web technologies"; Expert Systems with Applications 36(2), 3795–3804.

[Christensen and Schiaffino 2014] Christensen, I., Schiaffino, S.: "A Hybrid Approach for Group Profiling in Recommender Systems"; Journal of Universal Computer Science, 20, 507–533; 2014; `http://www.jucs.org/jucs_20_4/a_hybrid_approach_for`

[ELGG 2008] ELGG: "ELGG, a powerful open source networking engine", Available online: `http://www.elgg.org/`.

[Fernandez-Lopez et al. 1997] Fernández-López, M., Gómez-Pérez, A, Juristo, N.: "Methontology: From ontological art towards ontological engineering."; Symposium on Ontological Art Towards Ontological Engineering of AAAI, 33–40.

[Jaschke et al. 2000] Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: "Tag recommendations in folksonomies" 4702, 506–514.

[JavaBridge 2010] JavaBridge: "PHP/JavaBridge" Available online: `http://php-java-bridge.sourceforge.net`.

[Jena 2009] Jena: "Jena - Semantic Web Framework for Java", Available online:. `http://jena.sourceforge.net/`.

[Jung and Kazienko 2012] Jung, J., Kazienko, P.:"Advances on Social Network Applications", Journal of Universal Computer Science, 18; 2012; `http://www.jucs.org/jucs_18_4/advances_in_social_network`

[Kazienko et al. 2013] Kazienko, P., Lughofer, E., Trawinski, B.:"Hybrid and Ensemble Methods in Machine Learning"; Journal of Universal Computer Science, 19, 456–461; 2013; `http://www.jucs.org/jucs_19_4/hybrid_and_ensemble_methods`

[Pazzani and Billsus 2007] Pazzani, M., Billsus, D.: "Content-based recommendation systems"; Lecture Notes in Computer Science 4321, 325–341.

[Peis et al. 2008] Peis, E., del Castillo, J. M. M., Delgado-López, J. A.: "Semantic recommender systems. Analysis of the state of the topic", *Hipertext.net* **6**.

[Ricci et al. 2010] Ricci, F., Rokach, L., Shapira, B., Kantor, P. B. : "Recommender Systems Handbook", 1st edn, Springer-Verlag New York, Inc., New York, NY, USA.

[Sarwar et al 2001] Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: "Item-based collaborative filtering recommendation algorithms"; Proceedings of the 10th international conference on World Wide Web', WWW '01, ACM, New York, NY, USA, pp. 285–295.

[Schafer et al. 2007] Schafer, J.H.J.B., Frankowski, D., Sen, S.: "Collaborative filtering recommender systems"; Lecture Notes in Computer Science 4321, 291–324.

[Van der Wal 2009] Van der Wal, T.: "Understanding folksonomy: catalyzing users to enrich information", O'Reilly Media, Inc.

[W3C 2009] W3C: "OWL 2 Web Ontology Language Primer"

[Wang et al. 2000] Wang, R.-Q., Kong, F.-S.: "Semantic-enhanced personalized recommender systems", 12–25.

[Wozniak 2011] Wozniak, M.: "A hybrid decision tree training method using data streams"; Knowledge and Information Systems 29, 335–347.

[Yakut and Polat 2012] Yakut, I., Polat, H.: "Privacy-preserving hybrid collaborative filtering on cross distributed data'"; Knowledge and Information Systems 30, 405–433.