Evaluating Ontology-Based User Profiles

Silvia Calegari and Gabriella Pasi

Department of Informatics, Systems and Communication (DISCo),
University of Milano-Bicocca,
v.le Sarca 336/14, 20126 Milano (Italy)
{calegari,pasi}@disco.unimib.it

Abstract. User profiles play an important role in any process of personalization as they represent the user's interests and preferences. Only if a user profile faithfully represents the information related to a user a system may rely on it. This paper shortly presents a comparative evaluation between two distinct approaches that analyze textual documents for defining user profiles based on the usage of the YAGO general purpose ontology. The performed evaluations compare the two approaches both by the robust index measure and their efficiency.

Keywords: User profile, Ontology, Evaluation.

1 Introduction

The issue of personalization is becoming more and more important in various research domains. In fact, there is an increasing need to define personalized systems that tailor their outcomes to the users' context, to the aim of better complying to their expectations. A user profile plays a key role for the definition of personalized systems; it models several knowledge dimensions related to a user, such as his/her personal data, background knowledge, topical preferences, etc. The knowledge represented in a user profile is analyzed and then used to improve the standard behaviour of the considered system. A personalized system works well if the knowledge stored into a user profile represents at best the information related to a user.

In the literature several formal representations of user profiles have been proposed, such as sets of weighted keywords, semantic networks or hierarchies of concepts [4]. Ontologies have been recently considered as a valuable support to express a more structured and complete knowledge representation of user profiles. In fact, they allow to enrich the expressiveness of the information represented in a profile by using formal languages like RDFS or OWL. The existing models to build user profiles based on ontologies are mainly focused on approaches either relying on data mining techniques [6,10] or adopting external reference knowledge [3,8] to capture the meaning of the user's preferences.

In this paper our attention is on strategies that make use of ontologies (like external reference knowledge) to build user profiles. In particular, our approach

R. Huang et al. (Eds.): AMT 2012, LNCS 7669, pp. 377–386, 2012.

allows to generate ontological user profiles based on the general purpose ontology YAGO [9]. YAGO consists of several million of entities and facts, where a fact is a triple of two entities and the relation between them. In YAGO 99 relations have been defined. In this work, two strategies able to extract the meaningful entities and facts from YAGO are considered. The first strategy [1] is able to disambiguate the YAGO information acquired during the knowledge extraction process by combining the user's local knowledge (i.e., user's documents), and the user's global information (i.e., the YAGO ontology). The second strategy makes use of the query2YAGO query processor [5] that is aimed to search for YAGO facts according to a specific syntax. In this paper, we have extended query2YAGO in order to define a new methodology that allows to navigate and extract information from YAGO related to the user topical interests.

The comparative evaluations of the two above strategies include both qualitative evaluations and efficiency evaluations; the qualitative evaluations allow to analyze the obtained user profiles in terms of amount of noisy information gathered by the considered extraction process. The efficiency evaluations are finalised at testing the time required by each of the two strategies to build the user profiles.

The paper is organized as follows: Section 2 shortly introduces the considered strategies for the ontological user profiles definition, whereas in Section 3 the evaluations are presented that compare the effectiveness of the two methods. In Section 4 some conclusions are stated.

2 Building Ontological User Profiles

In this section the two considered strategies that make use of the YAGO ontology to express user's preferences in a semantically meaningful way are described. Both of them take in input a set of documents representing the user's preferences as well as the YAGO knowledge-base, and they generate in output a user profile constituted by the meaningful portions of YAGO related to the contents of the provided documents. Figure 1 shows an overview of the process undertaken by both strategies. The set of documents that are representative of the user's interests is indexed by a standard procedure. The output of the indexing procedure is a set of weighted keywords where the weights are computed by applying one of the classic weighting functions (e.g., the standard normalized Tf-Idf [7]). We call the selected weighted keywords the *interest-terms*. Then the two methods analyze the set of interest-terms to extract the YAGO sub-graphs as shown in Figure 1; the methods differ in the extraction process. Finally, the obtained YAGO knowledge portions are formally represented into the ontological language RDFS[1].

In Subsection 2.1 a short explanation of the YAGO ontology is provided, whereas in Subsections 2.2 and 2.3 the two methodologies for the user profile extraction are described.

[1] `http://www.w3.org/TR/PR-rdf-schema`

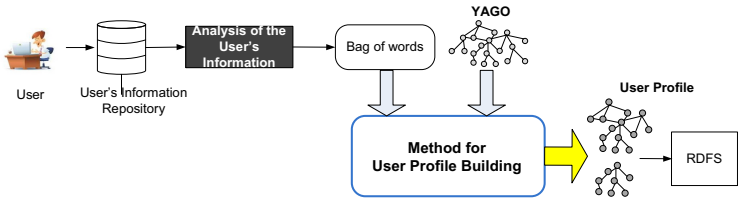

Fig. 1. Overview of the general process to build a user profile

2.1 The YAGO Ontology in a Nutshell

YAGO is one of the largest knowledge bases actually available, and it is composed of entities and facts. Currently, the YAGO knowledge base contains about 1.95 million entities and 19 million facts. The YAGO model has been defined as an extension of RDFS as explained in [9] where entities represent the objects in a world knowledge-base model. YAGO's authors have defined entities as *abstract ontological objects*; more specifically entities may be literals, or words, or classes, or relations, or fact identifiers. Entities that are neither fact identifiers nor relations are defined as *common entities*. Common entities that are not classes are called *individuals*. Entities constitute `arguments` of a fact, and a single fact is a triple constituted by two entities and the relation name linking them. An example of a fact is *(Mia Farrow, isMarriedTo, Frank Sinatra)*, with the meaning that *Mia Farrow has been married to Frank Sinatra*. Moreover, with each fact a *fact identifier* is associated to link, for example, URL information with the knowledge of an other fact. If the fact *(Mia Farrow, isMarriedTo, Frank Sinatra)* has the identifier *#1*, then it is possible to generate a new fact as *(#1, foundIn, http://en.wikipedia.org/wiki/Mia_Farrow)* to have more information on *Mia Farrow* and her marriage. Based on these notions, the YAGO model $\mathcal{M}_{YAGO}$ is defined as:

Definition 1 *$\mathcal{M}_{YAGO} = \langle E, \mathcal{F} \rangle$, where $E = \mathcal{I} \cup \mathcal{C} \cup \mathcal{R}$ where $\mathcal{I}$ is the set of fact identifiers, $\mathcal{C}$ is the set of common entities, $\mathcal{R}$ is the set of relation names, and $\mathcal{F}$ is the set of YAGO facts.*

Based on the application of the two strategies described in Sections 2.2 and 2.3, a user profile is defined as a subset of YAGO. Formally, a user profile is defined as $\mathcal{UP} = \langle E_{\mathcal{UP}}, \mathcal{F}_{\mathcal{UP}} \rangle$ such that $\mathcal{UP} \subseteq \mathcal{M}_{YAGO}$.

2.2 Building the User Profile: The First Strategy.

Figure 2 shows the main phases of the technique proposed in [1] to automatically build user profiles by using YAGO. Here below, a short explanation of each phase is reported.

***Common Entities Identification* Phase.** The objective is to discover the set of the YAGO common entities that are related to the set of interest-terms $\mathcal{IT}$

by string containment. Let *int* be an interest-term and c be a common entity, then *int* can be equal to c or it can be contained into c.

***Common Entities Disambiguation* Phase.** The objective is to reduce the noisy information gathered by the previous phase. This means to eliminate the YAGO common entities not related to the user topical interests by considering two types of information: local knowledge and global knowledge, respectively. The local knowledge on c determines its importance with respect to the set of interest-terms. Then the local knowledge weight of a common entity c, w^c_{LK}, is computed as an average of the weights that are associated with the interest-terms related to the common entity c. Instead, the global knowledge on c allows to explore its possible interpretations in YAGO. To this aim, we have considered the YAGO facts where common entities are linked by the *Type* (or *instance-of*) relation. The global knowledge weight of a common entity c, w^c_{GK}, is computed on the basis of how many other common entities share the same YAGO knowledge linked by the *Type* relation.

To associate an overall score with a common entity c, a linear combination of its weights w^c_{LK} and w^c_{GK} is applied as $w_c = \alpha * w^c_{LK} + (1 - \alpha) * w^c_{GK}$, where $0 \leq \alpha \leq 1$. The parameter α has been set to the value 0.6 in order to give a slightly higher importance to the local knowledge. A threshold value t is used to individuate the common entities that are representative of the user's interests.

***Rules for the Knowledge Extraction* Phase.** The objective is to extract the YAGO facts containing the set of common entities obtained as output of the previous phase by the definition of four specific rules. The YAGO entities constitute the *arguments* of a fact where $arg1$ identifies the YAGO entities that appear as first argument of a fact, whereas $arg2$ identifies the YAGO entities that appear as second argument of a fact. The four rules are: rule 1) only $arg1$ carries information useful to a common entity identification, rule 2) only $arg2$ carries information useful to a common entity identification, rule 3) both $arg1$ and $arg2$ together carry information useful for common entities, rule 4) either $arg1$ or $arg2$ may contain information useful to a common entity identification. Each of the 99 YAGO relations has been manually associated with the right rule.

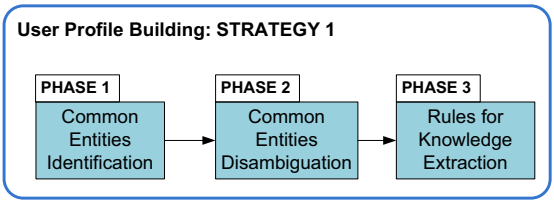

Fig. 2. Overview of the process to build a user profile

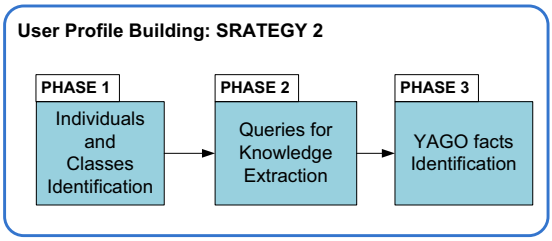

Fig. 3. Overview of the process to build a user profile

2.3 Building the User Profile: The Second Strategy

The second strategy we consider is proposed for the first time in this paper; it is able to semi-automatically extract portions of YAGO related to the set of interest-terms by using a simple query processor named *query2YAGO* [9]. The *query2YAGO* Java application has been defined by the developers of the Max-Plank Institute in order to search YAGO facts according to a specific query language. This query processor has been integrated in the NAGA semantic search engine [5] that can operate on knowledge-bases organized as graphs (like YAGO) in order to search the sub-graphs that match the user's query.

We have extended *query2YAGO* in order to search the YAGO sub-graphs related to the set of interest-terms as described in this section. Before providing the explanation of the phases devoted to the definition of the user profile, it is necessary to give a short explanation of the *query2YAGO* syntax as a technical documentation of the language is not available.

The syntax of a generic query is: $Q = e_1\ r\ e_2$, where $e_1, e_2 \in \mathcal{C} \cup \mathcal{I}$ and $r \in \mathcal{R}$, with the meaning of searching for facts containing e_1 as the first argument of a fact and e_2 as the second argument of a fact linked by the relation name r. During the query evaluation, the system automatically expands each argument (both e_1 and e_2) with its possible semantic interpretations in YAGO by analyzing the *Means* relation. For example, if $e_1 = guitar$ then the following query is implicitly defined: $Q = guitar\ means\ ?y$, where $?y$ individuates all the *meanings* of *guitar* in YAGO. Then, the output is the following set of common entities related to *guitar*: E_{guitar}={*Guitar_album, Guitar_song, Matt_Murphy_(blues_guitarist), wordnet_guitarist*}. Each common entity in E_{guitar} will substitute the original e_1 indicated by the user during the formulation of his/her query (the same happens for e_2). This way, if $|E_{e_1}| = M$ and $|E_{e_2}| = N$ then $M \times N$ queries will be defined and evaluated in order to extract all the YAGO facts related to e_1 and e_2 linked by the relation name r. When a user is interested in finding all the YAGO portions related to e_1 and e_2 without any constraint on the relation names linking them, it is possible to define the following query: $Q = e_1\ ?\ e_2$, where the character ? indicates all the YAGO relation names. Here below, an explanation of all the phases for building the user profile is reported.

***Individuals and Classes Identification* Phase.** From the set of interest-terms $\mathcal{IT}$, the user is asked to identify two sub-sets: (1) sub-set of individuals, and (2) sub-set of classes, respectively. To separate the interest-terms allows to formulate specific queries in *query2YAGO* to constrain the navigation of the YAGO knowledge-base in a proper way. Formally, the user judges $\mathcal{IT}$ to identify the sub-set of individuals $\mathcal{I}_{\mathcal{IT}}$ and the sub-set of classes $\mathcal{C}_{\mathcal{IT}}$ such that $\mathcal{IT} = \mathcal{I}_{\mathcal{IT}} \cup \mathcal{C}_{\mathcal{IT}}$ and $(\mathcal{I}_{\mathcal{IT}} \cap \mathcal{C}_{\mathcal{IT}}) = \emptyset$. This is the unique phase that requires the intervention of a user in the whole user profile building process.

***Queries for Knowledge Extraction* Phase.** The objective is to extract the YAGO sub-graphs related to the two sub-sets (i.e., $\mathcal{I}_{\mathcal{IT}}$ e $\mathcal{C}_{\mathcal{IT}}$) identified in the previous phase. To this aim, the *query2YAGO* query processor has been extended in order to manage four types of queries. These queries can be logically divided into two groups: (1) the first two types of queries allow to extract the YAGO knowledge where facts contain relation names that directly link individuals and classes in $\mathcal{IT}$, and (2) the last two types of queries allow to select additional facts related to individuals and classes in $\mathcal{IT}$ by exploring the YAGO knowledge-base. An explanation of the four types of queries is given here below:

First Type: it selects the YAGO facts where a direct association between an individual $i_j \in \mathcal{I}_{\mathcal{IT}}$ and a class $c_k \in \mathcal{C}_{\mathcal{IT}}$ exists. At this phase, two types of queries have to be defined in order to seek for individuals (or classes) appearing either as the first argument of a fact or as the second argument of a fact. Thus, two types of queries are defined as $Q_1 = i_j\ ?\ c_k$ and $Q_1 = c_k\ ?\ i_j$.

Second Type: it selects the YAGO facts containing two individuals $i_j, i_s \in \mathcal{I}_{\mathcal{IT}}$ where $i_j \neq i_s$. The query is defined as $Q_1 = i_j\ ?\ i_s$.

Third Type: it selects the YAGO facts where an individual $i_j \in \mathcal{I}_{\mathcal{IT}}$ shares the same knowledge of two classes $c_k, c_h \in \mathcal{C}_{\mathcal{IT}}$ where $c_k \neq c_h$. The aim is to extract from YAGO additional information related to individuals and classes in $\mathcal{IT}$. To this aim a sequence of the following queries: $Q_3 = i_j\ ?\ ?x$; $c_k\ ?\ ?x$; $c_h\ ?\ ?x$ is defined.

At this step, the main problem is to disambiguate the YAGO information obtained during the evaluation of the above sequence of queries. In fact, as previously reported, each interest-term (like an individual or a class) is automatically expanded with its possible meanings in YAGO by the *Means* relation. So that, if a user is interested in the sport tennis and *Agassi* $\in \mathcal{I}_{\mathcal{IT}}$, then it can be substituted with the two following YAGO individuals i.e., *Andre Agassi* and *Carlos Agassi*, but only the first one can satisfy the user's interests as, in this example, the user has preferences on the *tennis* topic. In order to select only facts related to the user's interests and preferences, this type of query searches YAGO facts where the knowledge associated with an individual is the same also for the two classes. We consider two classes because the use of a unique class is not sufficient for extracting non relevant facts; in fact, as it happens for an individual, the YAGO-interpretations of a class can be also ambiguous. By adopting

the knowledge of an individual plus the knowledge of two classes can reduce the possibility to discover noisy information from YAGO.

Fourth Type: it selects the YAGO facts where three classes $c_w, c_k, c_h \in \mathcal{C}_{\mathcal{IT}}$ with $c_w \neq c_k \neq c_h$ share the same knowledge. To this aim the following queries are generated: $Q_4 = c_w\ ?\ ?x;\ c_k\ ?\ ?x;\ c_h\ ?\ ?x$.
The idea underlying this type of query is the same of the previous one; the objective is to extract additional knowledge from YAGO related to the classes in $\mathcal{C}_{\mathcal{IT}}$. We consider three classes in order to minimize the number of non-relevant YAGO facts with respect to the user's expectations.

***YAGO facts Identification* Phase.** The outputs of the execution of the four types of queries are sub-portions of the YAGO knowledge related to the analyzed set of interest-terms $\mathcal{IT}$. Unfortunately, the output provided by *query2YAGO* is not conform to the YAGO fact syntax (i.e., a YAGO fact is a triple between two common entities and the relation name linking them). For example, if *Andre Agassi* $\in \mathcal{I}_{\mathcal{IT}}$ and *player* $\in \mathcal{C}_{\mathcal{IT}}$, then the result of the following query Q_1 = "*Andre Agassi*" ? *player* is RESULT=*"?=type, ?"Andre Agassi"=Andre_Agassi, ?player=wordnet_player"*. Thus, a parsing is needed in order to redefine, for example, the previous result in the standard YAGO fact syntax i.e., *(Andre Agassi, Type, wordnet player)*. To reconduct the set of results obtained by the *Queries for Knowledge Extraction* phase in the YAGO triples allows to convert them in the ontological language RDFS by using predefined scripts.The *YAGO facts Identification* step is then divided into two sub-phases: (1) to analyze each result obtained by the previous phase in order to modify it according to the YAGO fact syntax, and (2) to remove duplicated YAGO facts from the set of facts produced by sub-phase (1).

3 Evaluations

Evaluations have been performed to compare the strategies of Section 2 to both assess the quality of the knowledge gathered in the user profiles, and analyze their behaviour in terms of efficiency. Qualitative evaluations test the quality of the information stored in the user profiles in terms of ambiguous knowledge gathered.

To evaluate the two strategies, we have asked ten users to collect the documents that are more representative of their interests. The ten main user's collections (made up of 100 documents each) are representative of the following topics: *architecture, astronomy, botany, cuisine, health and fitness, literature, music, tennis, travel* and *wine*. The topics selected by the users are related to a broad spectrum of knowledge, and this allows to test the effectiveness of the two methodologies in different areas of interest.

The ten users had also a significant role during the qualitative evaluations; they have acted as assessors by evaluating the quality of the twenty sets of common entities defined in the user profiles obtained with the two applied methodologies on the ten topics. In fact, for each set of common entities obtained by the application of the two strategies of Section 2, each user has expressed a judgement on each common entity by classifying it into two distinct groups: a set of

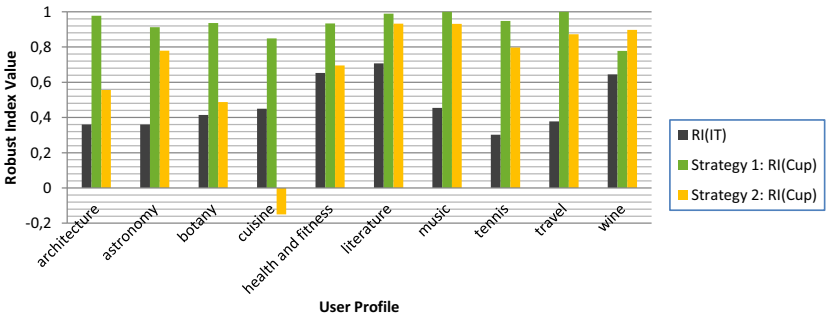

Fig. 4. RI metric evaluations for all the user profiles

positive common entities, and a set of uncorrelated common entities (i.e., not in line with the user's interests).
In Subsection 3.1 the qualitative evaluations are presented, whereas in Subsection 3.2 the efficiency evaluations are reported.

3.1 Qualitative Evaluations

To perform a qualitative evaluation of a user profile, we have adopted a simple measure called the robustness index (RI) [2]. In this paper, we adapt the RI metric to assess the quality of the set of common entities $\mathcal{C}_\mathcal{U}$ obtained as the outcome of the presented knowledge extraction processes. In fact, the extracted set of common entities $\mathcal{C}_\mathcal{U}$ may contain some noisy information, the amount of which we want to evaluate. We assume then that $\mathcal{C}_\mathcal{U}$ consists of two subsets, $\mathcal{C}_\mathcal{U}^+$ and $\mathcal{C}_\mathcal{U}^-$, $\mathcal{C}_\mathcal{U} = \mathcal{C}_\mathcal{U}^+ \cup \mathcal{C}_\mathcal{U}^-$, where $\mathcal{C}_\mathcal{U}^+$ identifies the positive common entities, and $\mathcal{C}_\mathcal{U}^-$ identifies the uncorrelated common entities. A common entity is identified as positive when it is semantically correlated to the considered user topical interest; on the contrary, a common entity is identified as uncorrelated when it is out of topic with respect to the considered user interest. The RI metric is defined as $RI(\mathcal{C}_\mathcal{U}) = \frac{|\mathcal{C}_\mathcal{U}^+| - |\mathcal{C}_\mathcal{U}^-|}{|\mathcal{C}_\mathcal{U}|}$, where $(|\mathcal{C}_\mathcal{U}^+| + |\mathcal{C}_\mathcal{U}^-|) = |\mathcal{C}_\mathcal{U}|$ and $-1 \leq RI(\mathcal{C}_\mathcal{U}) \leq 1$. Clearly, $\mathcal{C}_\mathcal{U} = 1$ if all common entities are classified as positive, while $\mathcal{C}_\mathcal{U} = -1$ if all common entities are classified as uncorrelated.

Experiments. As the input of the proposed methodologies is constituted by a set of interest-terms related to the user topical interests, our intuition is that if the number of uncorrelated interest-terms in $\mathcal{IT}$ is high, it can be more difficult to extract from $\mathcal{M}_{YAGO}$ the knowledge related to the right user topical interests. Figure 4 reports the values for $RI(\mathcal{IT})$, and $RI(\mathcal{C}_\mathcal{U})$ for both the two presented strategies over all user profiles. High RI values (i.e., closer to 1) are obtained when a few uncorrelated information is acquired by the considered knowledge extraction strategy. If there is a high number of uncorrelated interest-terms, it is plausible to assume that also a high number of uncorrelated common entities will be obtained by the extraction process. Both the two strategies for user profile building allow to obtain positive RI values by analyzing the set $\mathcal{IT}$. This

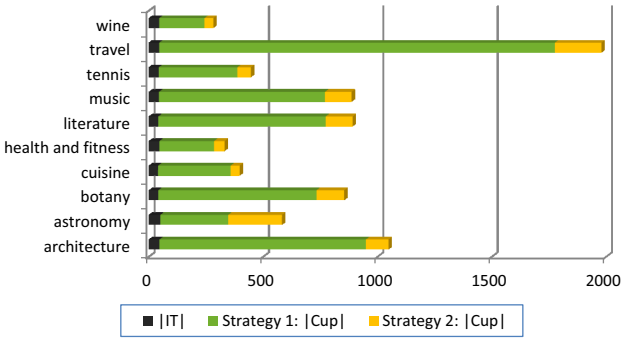

Fig. 5. Analysis of the number of common entities

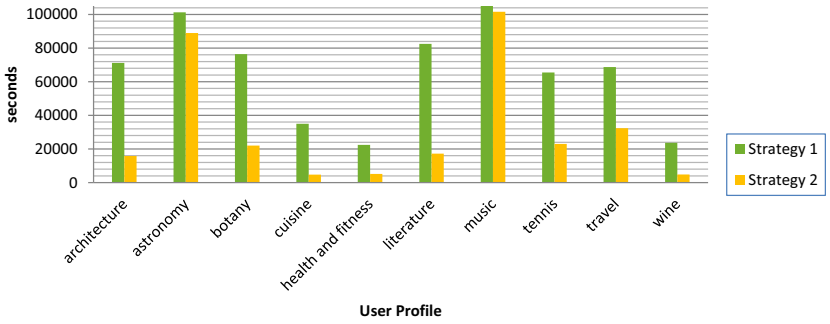

Fig. 6. Efficiency evaluations

indicates that they are able to select non-ambiguous knowledge in spite of the presence of noisy knowledge as starting point of their knowledge extraction process. By comparing the behaviour of the two strategies, it emerges that the first strategy outperforms the second one in all the user profiles but the user topical interest on *wine*. This means that the *Strategy 1* allows to better disambiguate the YAGO knowledge portions during the several phases of the process thanks to the analysis of the user local information that acts as an additional indicator of the knowledge from YAGO that is related to the user's interests. The good results obtained by *Strategy 1* in terms of robust index are also reinforced by analyzing the amount of common entities inserted in each user profile, as shown in Figure 5. In fact, not only *Strategy 1* extracts less uncorrelated knowledge from YAGO than *Strategy 2*, but it allows to obtain more information related to each user topical interest.

3.2 Efficiency Evaluations

The two considered strategies are now analyzed in order to consider the execution time necessary to obtain the YAGO sub-graphs. Figure 6 shows that for most user profiles *Strategy 2* outperforms *Strategy 1* (i.e., less execution time is used).

Similar timings are instead obtained for the user topical interests on *astronomy* and *music*; this happens because in YAGO some topics cover a huge amount of facts with respect to other topics (like *cuisine*), and then it is possible that the two strategies could manage the same amount of YAGO information before extracting the relevant one.
In general, the process of knowledge extraction of *Strategy 2* works faster than *Strategy 1*, but it exhibits a worst behaviour in terms of the robust index if compared with *Strategy 1*.

4 Conclusions and Future Works

In this paper both qualitative and efficiency evaluations have been presented to compare two distinct approaches that make use of the YAGO ontology to extract and represent user profiles from a set of textual documents representing users' interests. The evaluations have outlined some interesting results: the first considered strategy is better in selecting non ambiguous information, whereas the second strategy is more efficient than the first one.

In future works we will also compare the ontological user profiles defined by the strategies analysed in this paper with other methodologies presented in the literature that are able to build user profiles represented as ontologies.

References

1. Calegari, S., Pasi, G.: Personal ontologies: generation of user profiles based on the YAGO ontology. Information Processing & Management (2012) (in Printing)
2. Carpineto, C., Romano, G.: A Survey of Automatic Query Expansion in Information Retrieval. ACM Comput. Surv. 44(1), 1:1–1:50 (2012)
3. Daoud, M., Tamine, L., Boughanem, M.: A Personalized Graph-Based Document Ranking Model Using a Semantic User Profile. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 171–182. Springer, Heidelberg (2010)
4. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User Profiles for Personalized Information Access. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web. LNCS, vol. 4321, pp. 54–89. Springer, Heidelberg (2007)
5. Kasneci, G., Suchanek, F.M., Ifrim, G., Ramanath, M., Weikum, G.: NAGA: Searching and ranking knowledge. In: ICDE, pp. 953–962. IEEE (2008)
6. Li, Y., Zhong, N.: Mining ontology for automatically acquiring web user information needs. IEEE Trans. Knowl. Data Eng. 18(4), 554–568 (2006)
7. Robertson, S.: Understanding inverse document frequency: on theoretical arguments for idf. Journal of Documentation 60(5), 503–520 (2004)
8. Speretta, M., Gauch, S.: Miology: A web application for organizing personal domain ontologies. In: International Conference on Information, Process, and Knowledge Management, eKNOW 2009, pp. 159–161 (February 2009)
9. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A large ontology from wikipedia and wordnet. Journal of Web Semantic 6(3), 203–217 (2008)
10. Tao, X., Li, Y., Zhong, N.: A personalized ontology model for web information gathering. IEEE Trans. on Knowl. and Data Eng. 23, 496–511 (2011)