On the complexity of agent-mediated shared conceptualizations*

Gonzalo A. Aranda-Corral[1], Joaquín Borrego Díaz[2], and Jesús Giráldez-Cru[3]

[1] Universidad de Huelva. Department of Information Technology.
Crta. Palos de La Frontera s/n. 21819 Palos de La Frontera. Spain
[2] Universidad de Sevilla. Department of Computer Science and Artificial Intelligence.
Avda. Reina Mercedes s/n. 41012 Sevilla. Spain
[3] Artificial Intelligence Research Institute (IIIA-CSIC)
Campus Universidad Autónoma de Barcelona, Barcelona, Spain

Abstract. In the Social Web, folksonomies and other similar knowledge organization techniques may suffer limitations due to both different users' tagging behaviours and semantic heterogeneity. In order to estimate how a social tagging network organizes its resources, focusing on sharing (implicit) conceptual schemes, an agent-based reconciliation knowledge system based on Formal Concept Analysis is applied. This article describes experiments that focus on conceptual structures of the reconciliation process as applied to Delicious bookmarking service. Results will show the prevalence of sharing tagged resources in order to be used by other users as recommendations.

1 Introduction

The availability of powerful technologies for sharing information among users (social network members) empowers the organization of social resources. Among them, collaborative tagging represents a very useful process for users that aim to add metadata to documents, objects or, even, urls. Particularly impressive is the success of Delicious bookmarking service (`http://www.delicious.com/`) which is one of most popular social tagging platforms for storing, sharing, and discovering bookmarks.

As with other social behaviours, tagging shows advantages but also deficiencies, e.g. semantic heterogeneity. Projects like *Faviki* (`http://www.faviki.com`) or CommonTag (`http://commontag.org`) attempt to resolve these deficiencies. Within the network, and also based on user preferences, different tagging behaviours exist that actually obstruct automated interoperability. Although solutions exist that assist the user's folksonomy (tag clouds, tools based on related tag ideas, collective intelligence methods, data mining, etc.), personal organization of information leads to implicit logical conditions that often differ from

* Supported by TIN2009-09492 project of Spanish Ministry of Science and Innovation and *Excellence project* TIC-6064 of *Junta de Andalucía* cofinanced with FEDER founds.

the global interpretation of these conditions. Tagging provides a manner of weak organization for information that, although useful, is mediated by the individual user's behaviour. In order to show semantic heterogeneity, Formal Concept Analysis (FCA) [6] could be a sound tool . FCA is a mathematical theory that, applied to tagging systems, results in explicit sets of concepts that users manage by tagging, thereby organizing information into structured relationships.

As is argued in [7], tagging is essentially about sensemaking, a process where information is categorized, labeled and, most importantly, through which meaning emerges [9]. Even in a personal tagging structure, concept boundaries and categories are vague, so some items can be doubtfully labeled. Furthermore users also use tagging task for their own benefit, but nevertheless they contribute usefully to the public good [7]. Therefore, it seems it can be an interesting way to apply concept mining technologies to facilitate semantic interoperability among different tagging sets. Attention on semantical reconciliationg for different tag's sets lies in the fact that, since the user's tagging reflects their own set of concepts about documents, the tag-driven navigation among different resources could be insufficient due to semantic heterogeneity. Thus, to ensure an efficient use of another user's tag sets, some thought must be given to tags in order to achieve some consensus (also using FCA based tools), which allows us to navigate between different conceptual structures. In this scenario, it could be very important to attempt to delegate these tasks to intelligent agents. In [3], an agent-based knowledge conciliation method is presented.

The aim of this paper is to show how a Multiagent System (MAS) can be applied to shape the complexity of users' conceptual structures into a social bookmarking service, by comparing the *resource sharing* relationship amongst users against the *tagging sharing* relationship between users. The first relationship comprises a complex network where semantic similarities could be weak, while one expects that the second allows us some understanding about semantic interoperability based on tags and achieved by conciliation. The paper aims to shown the prevalence of semantic similarity (knowledge conciliation) in *tagging sharing* relation.

The following paper is organized as follows. Section 2 is devoted to the introduction of FCA. Section 3 reviews original agent-based reconciliation, which is applied in this paper. Section 4 describes the relational structure of tagging in Delicious. Sect. 5 provides a specific implementation of knowledge reconciliation. Section 6 present the experiments and some results. Finally, Sect. 7 discusses some conclusions.

2 Formal Concept Analisys

Convergence between the Social Web and the Semantic Web depends on the specific management of ontologies and similar knowledge organization tools.For example, Ontologies and tags/folksonomies must be reconciled in these kinds of projects. A useful bridge between these two kinds of knowledge representation could be *Formal Concept Analysis* [6], which provides semantic features for folksonomies.

According to Wille, FCA mathematizes a philosophical understanding of a concept as a unit of thought, composed by a set of objects (*extent*), which hold a common set of attributes, and a set of attributes (*intent*), which are hold for a set of objects. FCA also allows us to compute concept hierarchies from data tables.

The procedure to transform data into structured information, by means of FCA, starts from an entity called *Formal Context*. This formal context is a tupla $M = (O, A, I)$ composed by two sets, O (objects) and A (attributes), and a relation $I \subseteq O \times A$. We also can define an operator for one subset of any of them ($X \subseteq O$ or $Y \subseteq A$), named derivative operator, such as

$$X' := \{a \in A \mid oIa \text{ for all } o \in X\}, \quad Y' := \{o \in O \mid oIa \text{ for all } a \in Y\}$$

From this, a definition of (formal) concept can be obtained as a pair (X, Y) which holds $X' = Y$ and $Y' = X$. If we define the subconcept relation, $C_1 \subseteq C_2$ if $O_1 \subseteq O_2$, a hierarchy among concepts can be obtained and represented as a lattice.

Finally, logical expressions in FCA are *implications between attributes*, a pair of sets of attributes, written as $Y_1 \rightarrow Y_2$. This expression holds in M if all $o \in O$, its derivative set, $\{o\}'$ models $Y_1 \rightarrow Y_2$, and it is said that, $Y_1 \rightarrow Y_2$ is *an implication* of M. A set of implications, $\mathcal{L}$, is a (implicational) basis for M, if $\mathcal{L}$ is complete and non-redundant. FCA also defines a method to calculate an implication basis [6], which is called Stem Basis (SB). Its important to note that Stem Basis is only a particular case of an implication basis for context. Any other implication basis could be used as well.

SB will be used as set of rules in production systems for reasoning (as in [3]). A rule's support can be defined as the number of objects that contain all attributes Y_1 and they hold the implication too. Based on this property, a variant of implicational basis is defined, called Stem Kernel basis (SKB), composed by SB's subset where support of each rule is greater than zero.

To illustrate these three entities -formal context, concept lattice, and Stem Basis- an example based on a living being is depicted in fig. 1, up, left, and right, respectively.

2.1 Tagging, contexts and concepts

There are several limitations to collaborative tagging in sites such as Delicious. The first is that a tag can be used to refer to different concepts, i.e. there is a context dependent feature of the tag associated with the user. This dependence -called "Context Dependent Knowledge Heterogeneity" (CDKH)- limits both the effectiveness and soundness of collaborative tagging. The second is the Classical Ambiguity (CA) of terms, inherited from natural language and/or the consideration of different "basic levels" among users [7]. CA would not be critical when users work with urls (content of url induces, in fact, a disambiguation of

A	B	C	D	E
	Need water	Aquatic	Mobility	Legs
Cat	X		X	X
Leech	X	X	X	
Frog	X	X	X	X
Corn	X			
Fish	X	X	X	

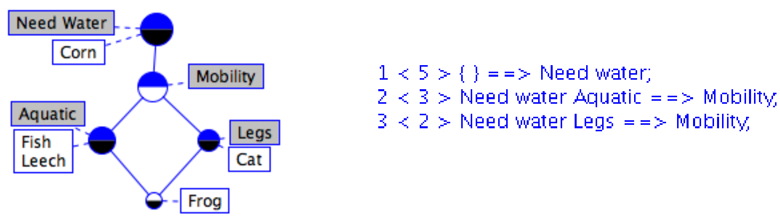

Fig. 1. Formal context and associated concept lattice and Stem Basis

terms because of its specific topic). In this case, the contextualization of tags in a graph structure (by means of clustering analysis) distinguishes the different terms associated with the same tag [5]. However, CDKH is associated with concept structures that users do not represent in the system, but that FCA can extract. Thus, navigation among concept structures of different users faced with CDKH. So the use of tagged resources for automatic recommendation is not advisable without some kind of semantic analysis. More interesting is the idea of deciphering the knowledge that is hidden in user tagging to understand their tagging behaviour and its implied meaning. In sites such as Delicious, CDKH is the main problem, because tags perform several functions as bookmarks [7].

3 Agent-based Reconciliation

Users's Knowledge Conciliation aims to exploit an important benefit of the Web 2.0, namely information and knowledge sharing. A potential threat is that semantic techniques are adapted to each user. Over time, the user's knowledge can vary a great deal, and this difference could create knowledge incompatibility issues. To navigate through the set of tags and documents from different users, process can be delegated to agents, in order not only to make these different conceptualizations compatible even to scale the process to a great number of users. The agent-based reconciliation process was succesfully applied in Mobile Web 2.0 [2] A agent-based conciliation algorithm was presented in [3]. It is based on the idea that conceptual structure associated with tags gives more information about the user's tagging. The algorithm runs in six steps (see Fig. 2):
1.- Agent creation: It starts creating two Jade[4] agents, passing through agent names and SinNet data as parameters.
2.- Each agent then builds its own formal contexts and Stem basis.

[4] `http://jade.tilab.com`

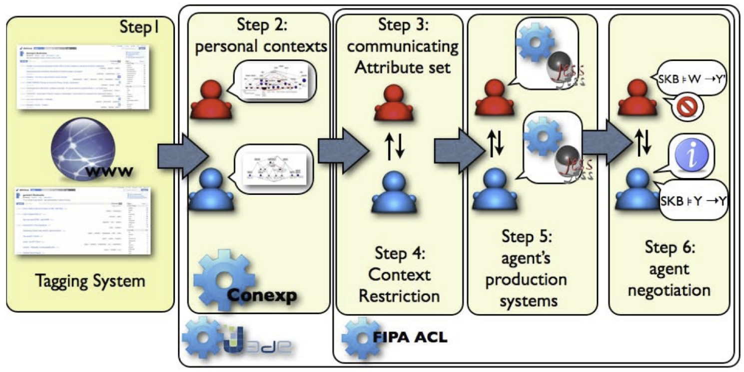

Fig. 2. Basic Knowledge Conciliation Algorithm

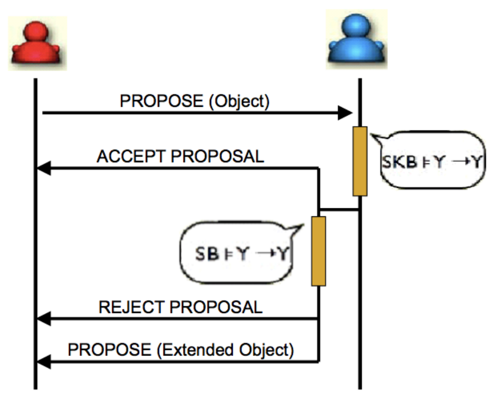

Fig. 3. Agent negotiation

3.- Initializing dialogue step: The agent executes tasks related to communications: It sends its own language (attribute set) to the other agent, and also prepares itself to receive the same kind of messages from the other agent.
4.- Restrictions of formal contexts: After this brief communication, each agent creates a new (reduced) set of common attributes, and with them a new context to which are added all of the objects from the original context, along with the values and attributes of the common language.
5.- Extraction of the production system (Stem Basis) for the new contexts.
6 .-Knowledge negotiation between agents (Fig. 3): Agents establish a conversation based on objects, accepting them (or not) according to their tag set and their own Stem Kernel Basis: if the object matches the rules, it is accepted, if not the production system is applied, considering the object's tags as facts, getting the answer (new facts which should be added in order to be accepted as a valid object) that is added to the object and re-sent to the other agent to be accepted.

Once this process is completed, the agents will achieve a common context. So they can extract new concepts and suggestions from a common context, and therefore, a shared conceptualization.

4 Delicious bookmarking service

We have chosen the bookmarking service Delicious due to its large volume of data. In Delicious, objects are web links (urls), and attributes are tags. Users save their personal web links tagged with their personal tags. But several users may share common objects (with different attributes for each one), or common attributes (tagged in different links). The structure and dynamics of tagging with Delicious have been extensively analyzed [7]. Because of limited computing capacity, certain reduction operations must be performed in order to ensure the normal functioning of the solution presented in this paper. Therefore, a subset of public Delicious data has been extracted, in which all the links are tagged with the tag *haskell*, and saved in a private database (DB) used to drive experiments. We have selected this tag (*haskell*) in order to reduce the possible semantic hetereogenety generated by other more general tags. In this case, this tag can be almost always found related to the functional programming language with the same name. Hence, this hetereogeneity is considerably reduced.

The process of obtaining this data is achieved through a query by tag (*haskell*) in Delicious. The result of this query is a list of links and its public information: user who tagged it and the tags tagged in that link by that user. All this content is saved in our private DB, generating a huge amount of information: lists of links, users, tags, and tuples {*user, link, tag*}. This DB initially has 4327 users, 3163 links, 2715 tags and 57497 tuples. Data extraction was performed on March 1st, 2011. This data set has a volume large enough to expect significant results. However, this set of data does not encompass all the links related to the *haskell* tag, instead only the first query results.

4.1 Cleaning Delicious dataset

The Delicious extraction process above described, generates a large amount of information which is saved into our private DB. However, some optimizations can be performed in order to obtain a better organisation of data. Some of these tasks remove irrelevant or duplicated data contained in DB. Others improve relational structure of tags.

Removing irrelevant tags On the one hand, *haskell* is an irrelevant tag because all links are tagged with. So, it does not provide any useful information. On the other hand, there are several tags that appear only once after removing the tag *haskell*. Based on this, tags can be considered as marginals because they have no relations[5] with any other.

[5] A pair of tags are related when both of them tag the same link by the same user.

In order to ensure database consistency, besides removing such tags, all tuples containing these tags must be also removed. Moreover, links with no tags (after tuples elimination) have no sense in our experiments. Therefore, they are also removed. This task removes 45 tags, 101 links and 10550 tuples.

Removing equal links It is important to understand how Delicious stores urls to appreciate that some links can be repeated in different registers of database. For instance, if several users save the same url with different descriptions, Delicious creates as registers in its DB as differents descriptions, although the url is always the same.

For our purpose, this means an important problem of inconsistency in the set of objects (links). To solve it, repeated links are simplified in one of them, and tuples containg these links are updated. Additionally, if some tuples become repeated after updated, they are also removed. The result of this task is the elimination of 14 links and 3 tuples, and the updating of 552 tuples.

Removing equivalent links In the same manner of the previous case, several links can be duplicated in Delicious DB due to the users' behaviours when saving it, e.g., link with and without a final slash. As mentioned above, equivalent links are simplified, and tuples containing these links are updated. The result of this task is the elimination of 19 links and 40 repeated tuples, and the updating of 4086 tuples.

Joining singular and plural tags As is explained in section 2.1, there exists a limitation of extracting knowledge from the tags due to the own limitations of the users' natural language. In the next section, it is shown that tags present a clear relational structure. However, this structure can be strongly improved with few changes in the context. One of the more common examples is the usage of tags either in singular or in plural, depending on the user. This intersection brings about a separation that FCA tools can rarely correct. Nevertheless, some simple rules can be performed to drop these limitations and get a better running of the experiments.

Motivated by this argument, singular and plural tags are simplified in a unique tag. The process of obtaining these pairs of tags is quite simple: the same tag with and without a final *'s'*. But this rule must be considered with caution, because it is neither correct, nor complete. Consequently, not all the pairs singular-plural are found, and some incorrect pairs can be also found (in fact, 6 of the 197 results are incorrect and must be considered as exceptions to avoid the corruption of the data). However, the result of applying this rule is amazing: the elimination of 191 tags and 1667 repeated tuples, and the updating of 3765 tuples.

Joining equivalent tags Likewise, many tags can be grouped because they are very similar, but they are written with different words, e.g., *math, maths,* *mathematic, ...*. In the next section, the process of obtaning the main tags and the structure between them is ilustrated. In this case, no automatic process has been implemented for this task. However, the analysis of the structure of tags lets to know the most important groups of tags that must be clasified. They are *math, programming, functional programming, programming language, language, tutorial* and *toread*.

This task consists of simplifying these groups of tags, and updating the corresponding tuples. The result is the elimination of 52 tags and 158 repeated tuples, and the updating of 611 tuples.

Removing empty users and empty links Finally, a maintenance task must be performed in order to assure the coherence of the DB, i.e., removing users with no links tagged, and links with no tags. This taks removes 68 users and 1 link.

The resulting DB The resulting DB is composed by 4259, 3028, 2427 tags and 45079 tuples.

4.2 The relational structure of tags

In order to estimate the complexity of the relationships among tags of data source, a graph was generated, in which nodes appear as tags, which were interconnected by weighted edges, whose weight represents the amount of links commonly shared according to a Delicious user. To understand the structure of the graph and the number of relevant tags, some simplifications have to be made.

Fig. 4 shows data resulting from *semantic communities* computing (using the method [4]), which is a simplified graph. This graph shows 5 different communities, demonstrating that tags of a same community are very interconnected, unlike tags of different communities, which display little connectivity. In the graph, each node is characterized by its color (determining the community it belongs to), its size (scaled according to its degree), and by the width of its edges (scaled according to the weight of the edge). Finally, only the most relevant nodes (27) and edges (138) are shown - accordingly measured by their importance in terms of degree and weight, respectively.

5 Multiagent System

Our aim is to find a good strategy in order to apply the reconciliation algorithm presented above in Delicious. This algorithm allows us to calculate the reconciled knowledge. However, this algorithm requires high computational resources. Hence, choosing the right pairs of users to execute the algorithm, among the whole community, remains a problematic issue. In order to execute a solution

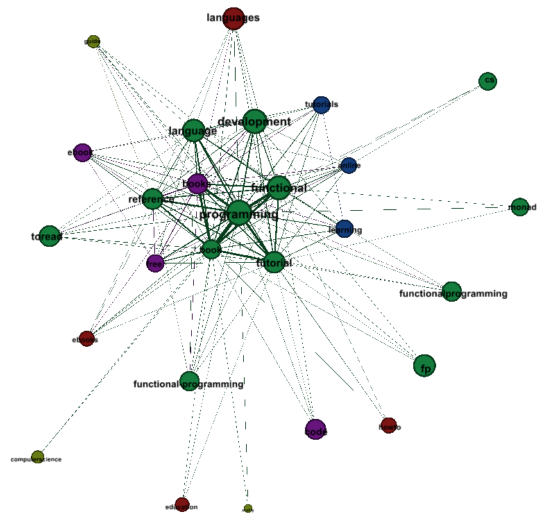

Fig. 4. Analysis of tag communities induced by *haskell* tag in Delicious (simplified)

that calculates reconciled knowledge for the whole tagging system, a negotiation based on MAS is proposed, in which agents represent tagging system users. They interact with each other to generate new common knowledge using the above mentioned algorithm. In the following section, the results we obtained are presented for different parameters used in the negotiation process. The MAS has also been implemented in Jade, where the implementation of the previous algorithm can be easily integrated. The execution of MAS can be described as the following steps:

1.- Initialization: In this step, as many user agents as needed are created. Only users sharing a minimum number of tags (threshold) participate in the MAS. Execution starts by creating an agent, called *control*, which passes this threshold as a parameter. This agent searches the DB for all pairs of users satisfying the threshold condition, and creates them within the MAS. Therefore, the agents present in the system are know as the *control* agent. The control agent may be useful to mange the MAS when integrated in more complex systems. Every $User_i$ must know its personal information (username, links and tags), and initialize itself by creating its own request queue. This queue contains references of all users having an equal or greater number of common attributes. It is sorted by the common attributes number, in descending order. Additional methods are equally needed to verify that a pair of users is only referenced in one of their request queues. Further experiments use the number of common objects of a pair of users as the threshold in order to compare results from both executions.

2.- Negotiation: User agents must execute a dual behaviour in order to perform the negotiation process: sending and receiving requests. This negotiation establishes a very simple method to decide when a pair of users starts the reconciliation process. Each user is only allowed to perform one reconciliation process at a time. Furthermore, received requests have priority over the sent ones. Two possible states for each user are defined: *free*, if it is not performing any reconciliation at the moment, and otherwise, *busy*. As such, only free users may send or receive requests. On one hand, every user sends proposals to the user having the highest priority in its request queue. If it receives a response, the reconciliation process with the addressee starts. Should this not be the case, it reiterates with the user having the next highest priority. On the other hand, every free user accepts any incoming proposals, even if it has already sent another proposal, which will be cancelled by timeout. The following conditions ensure that all of the conciliations will be processed: their number is finite, and there is always free users ready to accept new conciliations, reducing the number of unsolved processes. When starting a reconciliation, user's state switches from free to busy.

3.- Reconciliation: The algorithm presented in section 3 is used to calculate the common knowledge between two users. The steps 1 and 2 (user's concept lattice and SB) are executed only once, when the user runs it for the first time. The rest of the steps (3-6), are executed each time the user runs the algorithm. The obtained common knowledge, a formal context with objects and common attributes, is stored in the DB. Both users switch from busy to a free state.

4.- Finalization: When a user's request queue becomes empty, its behaviour is limited to receiving incoming proposals. However, if all the users' request queues are empty, no proposal is received by any of them. Therefore, this situation requires that the execution stop. The *control* agent is used to manage it. It is informed by every user when its request queue becomes empty. When all the users have completed this action, the control agent stops the MAS execution.

6 Experiments

Different experiments have been conducted with data described in section 4 using several criteria. The first criterion is setting a threshold of common attributes (tags) between users. The second criterion is setting a different threshold of common objects (urls). In both cases, the threshold is a necessary condition of a minimum number of attributes or objects that two users must have in common in order to execute the reconciliation algorithm. For each executed reconciliation process, a common knowledge is obtained. This knowledge is a formal context where the attributes are common to both users, and objects belong either to one of them, or both. In this way, the global result is a set of reconciled contexts.

The results obtained for both experiments using numerical and graphic representations are presented bellow. In order to do so, the results have been measured with five parameters for a fixed value of the threshold. They are the number of contexts obtained (there are as many contexts as number of executed reconciliation processes), and the average values of objects, attributes, concepts and implications per context. Finally, both experiments are compared.

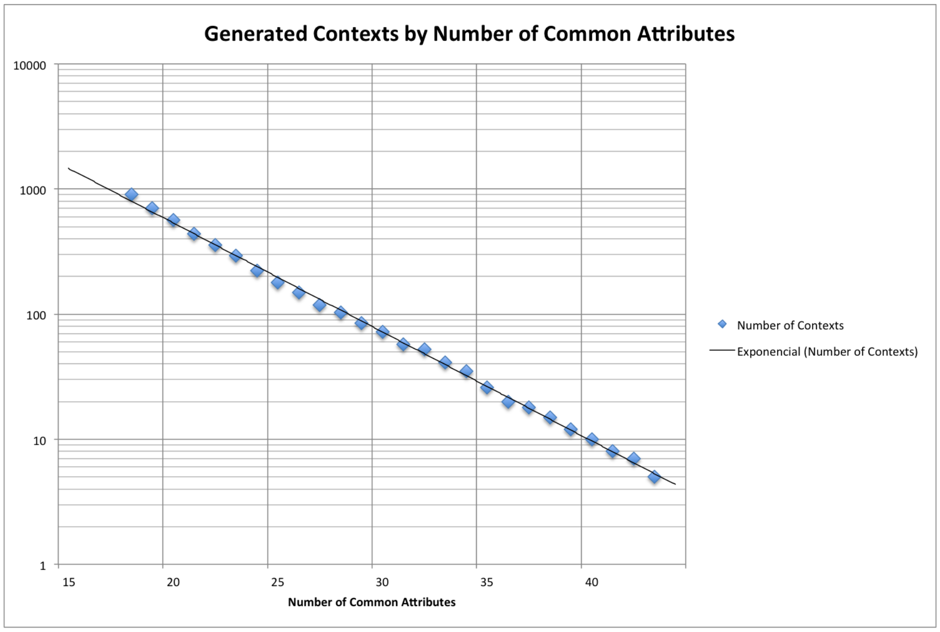

Fig. 5. Contexts generated by number of common attributes

Reconciliation from common attributes: In this experiment, the threshold value is set to 18. It implicates that two users having a language[6] size greater or equal to 18, reconcile their knowledge. It is assumed that users having a number of common attributes less than 18 do not share a relevant amount of information. In fig. 5, the graphics are plotted in logarithmic scale.

A total of 908 contexts were obtained, with an average value of 52.3 objects, 22 attributes, 8.1 concepts, and 22.8 implications per context. As the threshold value increases, the number of generated contexts decreases exponentially, as we can see in the log plot at figure 5. However, the four average values tend to increase. Although the number of contexts is smaller, they are semantically better, since the two users generating these contexts share more information. In this DB, the maximum number of common attributes is 64. In table 1, detailed results are shown for each threshold using this criterion.

Reconciliation from common objects: In the second experiment, the threshold value is set to 3. The implication is that two users having a set of common objects with size greater or equal than 3 reconcile their knowledge. As previously mentioned, it is assumed that sharing less than 3 objects is not relevant for the purpose of this study. In fig. 6, the results are represented. This case shows a total of 663 contexts, with an average value of 33.55 objects, 9.41

[6] The language between two users is the set of common tags that both of them use, independent of wether or not these tags have been used in different urls or not.

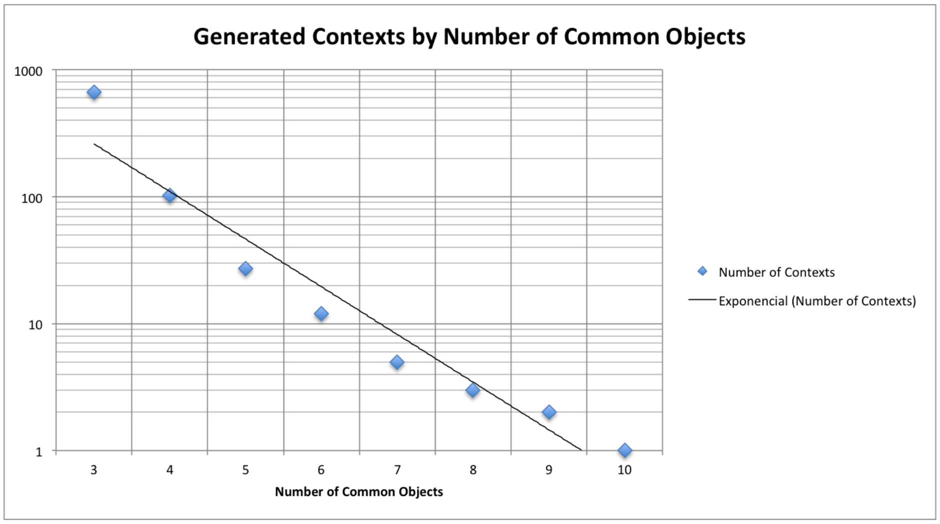

Fig. 6. Contexts generated by number of common objects

attributes, 6.09 concepts, and 9.75 implications per context. As the threshold value increases, the number of obtained contexts also has a high decrease. The maximum number of common objects is 11, which is very small. In table 2, we show the numerical result for this criterion depending on the threshold value.

6.1 Results

The results draw the conclusion that common attributes criterion is better than common objects criterion. On one hand, the decrease in generated contexts is higher when using common objects rather than common attributes. This is clearly proved in Fig. 5 and 6, where regression line has a higher slope in the case of objects, i.e., the decrease is higher using the objects criterion. On the other hand, the *semantic* validity of the generated contexts, measured along with their average values (i.e., objects, attributes, concepts and implications), is higher using attributes rather than objects. In the first case, average values increase linearly. It is then thought that the higher number of common attributes, the better reconciled context obtained. Unlike the first case, the second shows a constant function from a certain value of the number of common objects. It seems that the validity of the generated contexts does not depend on the number of common objects.

In conclusion, previous results lead us to think that the common attributes criterion separates more effectively the sample of generated contexts. Indeed, despite the fact that it returns a smaller amount of contexts, increasing the threshold value leads to results *semantically* better. Therefore, it is a good measurement of the semantic similarity of two users.

Threshold	18	19	20	21	22	23	24	25	26	27	28	29	30
Number of Contexts	908	701	562	439	358	292	223	178	148	118	103	85	72
Av. of Objects	52.3	54.7	57.5	60.0	53.6	65.8	69.8	74.2	75.9	78.6	80.1	83.7	85.8
Av. of Attributes	22.0	23.2	24.1	25.4	22.3	27.4	28.7	30.0	30.9	32.2	33.0	34.0	34.9
Av. of Concepts	8.1	8.5	8.9	9.5	8.9	11.0	11.7	12.8	13.3	14.0	14.6	15.3	16.0
Av. of Implications	22.8	24.4	25.9	27.6	25.7	31.5	34.0	36.3	38.4	40.4	41.6	43.4	44.9
Threshold	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**
Number of Contexts	57	52	41	35	26	20	18	15	12	10	8	7	5
Av. of Objects	89.7	92.0	93.0	95.5	98.8	101.8	104.7	109.3	113.3	115.9	117.6	120.9	131.6
Av. of Attributes	36.2	36.7	38.0	38.8	40.5	42.1	42.8	43.9	45.4	46.7	48.4	49.4	52.4
Av. of Concepts	17.0	17.2	16.7	18.2	19.2	20.7	22.4	23.8	24.7	25.9	26.6	27.6	30.6
Av. of Implications	47.8	48.3	49.1	51.5	54.6	57.3	59.7	62.4	65.6	68.7	72.3	75.7	85.6

Table 1. Results of the reconciliation process using common attributes.

Threshold	3	4	5	6	7	8	9	10
Number of Contexts	663	102	27	12	5	3	2	1
Av. of Objects	33.55	59.97	80.00	84.17	101.60	111.33	95.00	98.00
Av. of Attributes	9.41	15.90	23.41	22.67	37.60	42.67	37.50	37.00
Av. of Concepts	6.09	10.09	16.26	17.42	27.80	28.67	27.00	29.00
Av. of Implications	9.75	19.61	33.37	32.00	55.60	62.33	53.50	56.00

Table 2. Results of the reconciliation process using common objects.

7 Conclusions and Future Work

The experiments described in this paper show the prevalence of semantic techniques (tags) in resource sharing when users aim to exploit knowledge organization from other users in Delicious as a recommendation source. Although this result seems evident, Web 2.0 shows several examples where url sharing by social networks represent a powerful method for information diffusion (e.g. Twitter).

Therefore, we have empirical evidence that semantic similarity between users is better supported by using the method of reconciling the knowledge among users that have a large set of common attributes, rather than any other method. One of our lines of research is the intensive application of definability methods based on completion [1] in order to enrich the bookmarking system and to facilitate the reconciliation.

References

1. Alonso-Jiménez, J.A., Aranda-Corral, G.A., Borrego-Díaz, J., Fernández-Lebrón, M.M. and Hidalgo-Doblado, M.J. Extending Attribute Exploration by Means of Boolean Derivatives. Proc. 6th Int. Conf. on Concept Lattices and Their Applications. 2008. CEUR Workshops Proc. 433.

2. A. Aranda-Corral G., Borrego-DíazJ. and Giráldez-Cru J. Conceptual-Based Reasoning in Mobile Web 2.0 by means of Multiagent Systems. Knowledge Engineering Notes, Proc. 4th Int. Conf. Agents and Artificial Intelligence ICAART 2012, pp. 176-183, 2012
3. G. A. Aranda-Corral, J. Borrego-Díaz, Reconciling Knowledge in social tagging web services. Proc. 5th Int. Conf. Hybrid AI Systems (HAIS 2010), LNAI, vol. 6077. Springer-Verlag, Berlin, 383-390 (2010).
4. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, Vol. 2008, No. 10.
5. Ching Man Au Yeung and Nicholas Gibbins and Nigel Shadbolt, Contextualising Tags in Collaborative Tagging Systems,Proceedings of the 20th ACM Conference on Hypertext and Hypermedia. 2009.
6. Ganter, B., Wille, R.: Formal Concept Analysis - Mathematical Foundations. Springer, Heidelberg (1999).
7. Golder, S., Huberman, B.A. (2006), "The structure of collaborative tagging systems", Journal of Information Science, Vol. 32 No.2, pp.98-208.
8. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Discovering shared conceptualizations in folksonomies. Journal of Web Semantics 6(1), 38–53 (2008)
9. Weick, K. E., Sutcliffe K. M., Obstfeld, D. Organizing and the Process of Sensemaking, Organization Science, Vol. 16, No. 4, July-August 2005, pp. 409-421