HAL
archives-ouvertes.fr

SemTagP: Semantic Community Detection in Folksonomies

Guillaume Erétéo, Fabien Gandon, Michel Buffa

▶ To cite this version:

Guillaume Erétéo, Fabien Gandon, Michel Buffa. SemTagP: Semantic Community Detection in Folksonomies. 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Aug 2011, Lyon, France. 10.1109/WI-IAT.2011.98 . hal-01170978

HAL Id: hal-01170978
https://hal.inria.fr/hal-01170978

Submitted on 2 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SemTagP: Semantic Community Detection in Folksonomies

Guillaume Erétéo[1], Fabien Gandon[2], Michel Buffa[3]
[1] *Orange Labs, Sophia Antipolis, 06921 France*
guillaume@ereteo.net
[2] *INRIA - Edelweiss, 2004 rt des Lucioles, BP 93, 06902 Sophia Antipolis*
fabien.gandon@inria.fr
[3] *KEWI, I3S, University of Nice, France*
buffa@unice.fr

Abstract

Building on top of our results on semantic social network analysis, we present a community detection algorithm, SemTagP, that takes benefits of the semantic data that were captured while structuring the RDF graphs of social networks. SemTagP not only offers to detect but also to label communities by exploiting (in addition to the structure of the social graph) the tags used by people during the social tagging process as well as the semantic relations inferred between tags. Doing so, we are able to refine the partitioning of the social graph with semantic processing and to label the activity of detected communities. We tested and evaluated this algorithm on the social network built from Ph.D. theses funded by ADEME, the French Environment and Energy Management Agency. We showed how this approach allows us to detect and label communities of interest and control the precision of the labels.

1. Introduction and Related Works

Community detection helps understanding the distribution of actors and activities. Many tasks can benefit from the identification of communities of interests e.g. business intelligence, project team creation, technology monitoring, consulting, focused notifications in information systems, etc. Algorithms that tackle this problem are either hierarchical or based on heuristics [4]. Hierarchical algorithms produce a tree of community partitions by iteratively dividing the network into sub communities (top-down) or by merging communities into larger ones (bottom-up). Heuristics based algorithms, for instance random walk or analogies with electrical networks, exploit network characteristics to determine densely connected group of nodes. Among the heuristics based algorithms, the label propagation [11] (also known as RAK) proposes to detect communities by propagating labels in the social network as follows: (1) The algorithm assigns a unique random label to each node. (2) Each node *n* replaces its label by the label the most used by its adjacent nodes in the graph, if its own label is different. In case several labels are the most used, one is chosen randomly. (3) If at least one node changed its label, go to step 2. (4) Else nodes that share the same label form a community. Figure 1 presents this algorithm on a toy example.

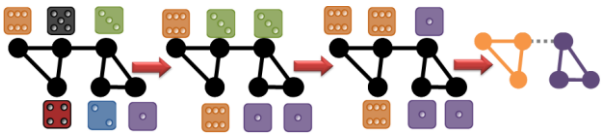

Figure 1. example of label propagation

Social web applications made social tagging popular: users categorize resources (e.g. media, blog posts, etc.) with freely chosen keywords called tags. This process generates a folksonomy: a set of actors describing a set of objects with a set of tags. A pioneering work by Peter Mika [8] investigated folksonomies as lightweight ontologies emerging from the usages of communities. Each tag may represent a community of interest that is composed of all the actors using this tag. Tags enable people to easily classify online resources for their personal use or for targeted communities, and to freely join online interactions. Tags shared by several users form a new source of links between users: "interaction produces similarity, while similarity produces interaction" [8]. For instance, during the Iran election, people overcame the media censorship with the Twitter social network by annotating their posts with the same tag, `#iranelection`, in order to interact and gather their information. Tags enable to link users and to label their emerging community. In [6] the authors improve community detection by

applying a clustering algorithm to a graph treating equally tags and resources.

Some tags are semantically related (hyponyms, synonyms, etc.) and a set of linked tags can also be viewed as a vocabulary shared by members of a community. Different approaches were proposed to structure folksonomies and identify semantic relations between tags with automatic processing or user contributions (see overview in [7]). Recently, [7] defined a method to combine automatic processing and manual user contributions to help online communities semantically enrich folksonomies and structure their own vocabularies. Once folksonomies are typed and structured, the relations between the tags and between tags and users provide a new source of affiliation networks, which enables us in this article to refine the labeling process of communities.

In this paper we propose to merge these three approaches (RAK, tag based labeling and folksonomy structuring) in order to perform community detections that take benefits, not only of the link structure of the social network, but also of the emerging semantics of folksonomies. We first introduce SemTagP, an algorithm that turns the RAK random label propagation into a semantic tag propagation in order to detect communities and meaningfully label them. Then we present how we implemented this algorithm with semantic web frameworks in order to take benefits of the ontological primitives used to type RDF graphs. Finally, we present the result that we obtained with a social network built from Ph.D. theses funded by the ADEME, the French Environment and Energy Management Agency.

2. SemTagP: Semantic Tag Propagation

SemTagP is an algorithm to detect and characterize communities from the directed typed graph formed by RDF descriptions of (social) networks and folksonomies. Using existing ontologies to represent online social networks [4], we can link and type online social networks, associate their actors to tags and semantically relate tags to each other.

SemTagP (Figure 2) is an extension of the RAK algorithm that turns the label propagation into a semantic propagation of tags: instead of assigning and propagating random labels, we assign to actors the tags they use and we propagate them using generalization relations between tags (e.g. `skos:narrower` / `skos:broader`) to merge over specialized communities and generalize their labels to common hyperonyms.

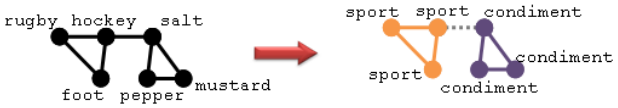

Figure 2. Semantic label propagation.

We use the directed modularity on RDF directed graphs [10] to assess the quality of the community partition obtained after each propagation loop. When a partitioned network has a high modularity, it means that there are more connections between nodes within each community than between nodes from different communities. More precisely, the modularity measures the fraction of edges within communities in the network minus the expected value of the same quantity in a network with the same community partition but with random connections between nodes [9] (the randomization of connections preserves the degree of the nodes). The modularity in a directed network is defined in [10] as follow:

Definition 1, directed modularity: let m be the number of edges of the network, A_{ij} the number of edges between i and j, c_i the community of i, $\delta(c_i,c_j) = 1$ if $c_i = c_j$, 0 otherwise, $d^{in}_{<i>}$ and $d^{out}_{<i>}$ the in-degree and out-degree of vertex i, the directed modularity is:

$$Q = \frac{1}{m}\sum_{i,j\in V}[A_{ij} - \frac{d^{out}_{<i>} d^{in}_{<j>}}{m}]\delta(c_i, c_j)$$

SemTagP iteratively propagates the tags in the network in order to get a new partitioning: nodes that share the same tag form a community. During a propagation loop each actor chooses the most used tag among its neighbors, for a tag t we count 1 occurrence for each neighbor using t and 1 occurrence for each neighbor using a `skos:narrower` tag of t. We iterate until the modularity stops increasing. The penultimate partitioned network is the output of the algorithm.

In our previous results on semantic social network analysis [5] we highlighted the importance of considering the diversity and the semantic of links between actors. Propagating tags through different types of relations, namely in different sub-networks, could produce different community partitions. Consequently, SemTagP is parametrized by the type of the analyzed relation. We formalize SemTagP as follow:

```
Algorithm SemTagP(RDFGraph network, Type relation)
1.  DO
2.    old_network = network
3.    //propagate tags (i.e. compute new partitions)
4.    FOR user in network.users
5.     user.tag=mostUsedNeighborTag(user, relation)
```

```
6.    END
7.  WHILE mod(network) > mod(oldNetwork)
8.  RETURN old_network
```

```
Algorithm mostUsedNeighborTags(User user,
Type relation)
1.  resultTag = null; max = 0
2.  tagTable = new hashTable()
3.  FOR agent in user.neigbors[relation]
4.    IF tagTable.exists(agent.tag)
5.      tagTable[agent.tag] ++
6.    ELSE
7.      tagTable[agent.tag] = 1
8.    IF(max < tagTable[agent.tag]){
9.      resultTag = agent.tag;
10.     max = tagTable[agent.tag]
11.   FOR broadTag in agent.tag.broaders
12.     IF tagTable.exists(broadTag)
13.       tagTable[broadTag] ++
14.     ELSE
15.       tagTable[broadTag] = 1
16.     IF max < tagTable[broadTag]
17.       resultTag = broadTag;
18.       max = tagTable[broadTag]
19.   END
20. END
21. RETURN resultTag
```

In our first experimentation, we witnessed that some tags with many `skos:narrower` relations absorbed too many tags during the propagation phase, such as the tag *environnement* (environment), which is ubiquitous in the corpus of the ADEME agency. Such tags grouped actors in very large communities. Consequently, we added an option to refine manually the results: after the first propagation loop we present the current community partition and labeling to a user that can reject the use of `skos:narrower` relations of tags labeling too large communities. Then, we restart the algorithm and repeat this process until no more relation is rejected, before completing the algorithm described above. For instance, during the partitioning of a social network with tags related to web topics, the user can reject `skos:narrower` relations of *web* such as *web* `skos:narrower` *semantic web*, in order to reveal the semantic web community.

We formalized here our algorithm. We will now see how we implemented this algorithm with the semantic graph engine KGRAM [1] that supports SPARQL 1.1 RDF query language. We delegate all the semantic processing performed on the graph to the semantic graph engine, taking benefits of SPARQL queries to exploit semantic relations between tags. Notice that the pattern matching mechanism of KGRAM's SPARQL implementation is based on graph homomorphism that is an NP complete problem. However, many optimizations enable us to significantly cut the time calculation of the RDF graph querying.

2.1 Semantic Tags Assignment

Different ontologies have been proposed to model folksonomies and social tagging activities and are used to generate RDF annotations. In particular, the SCOT[1] ontology provides "a consistent framework for expressing social tagging at a semantic level in machine-understandable way". Tagging ontologies identify tags with URIs and consequently turn these social labels into real objects (in the RDF sense) that can be semantically described. Thus we can leverage the meaning of these apparently flat labels by using them as the subject or the object of a triple. In particular, we can infer semantic relations between tags in order to structure the folksonomy with lightweight semantics. We infer semantics between tags, using the complete life-cycle proposed in [7], to enrich folksonomies by "combining automatic processing of tags and users' contributions through user-friendly interfaces". This cycle starts with a composite metric that combines several string-based metrics to reveal 3 main types of relations between tags: `skos:related`, `skos:closeMatch` and `skos:narrower`. Then users can validate, reject, or propose semantic relations through a web navigation tool, and emerging conflicts are solved by a referent user that maintains a consensual point of view. This cycle is iteratively restarted to maintain a folksonomy consensually augmented with semantic assertions (see [7] for more details).

We describe in the next section the way we use the resulting structured folksonomy to propagate tags, taking benefit of RDF typed graphs and SPARQL requests to ease the implementation of the different steps required by the algorithm.

2.2 Semantic Tag Propagation

The propagation step consists in iteratively assigning to each actor the most frequent tag among the actors he is linked to. In order to consider generalization relations between tags, we strengthen the score of a tag with the score of its `skos:narrower` tags. For instance, we exploit the semantic statement energy `skos:narrower` renewable energy by counting one more occurrence of the tag energy for each occurrence of the tag renewable energy.

We start each loop with a query that extracts for each actor the tags of its neighbors (for a given parameterized relation), their broader tags, and we order the results by actors and tags:

[1] http://scot-project.org/scot/spec/scot.html

```
1. select ?user ?tag ?y where {
2.  ?user param[rel] ?neighbor
3.  {{?neighbour scot:hasTag ?tag }
4.    UNION
5.    {?neighbour scot:hasTag ?tag2
6.     ?tag skos:narrower ?tag2
7.     filter(exists{?x scot:hasTag
 ?tag})}}
8. } order by ?user  ?tag
```

Different parts of the `mostUsedNeighboursTags()` function described above are encoded in this query:

- line 3 encodes the selection of the tag of a user's neighbors
- lines 5 to 7 encode the selection of a tag that is broader than the tag of a user 's neighbor
- line 8 orders the projections for each user and tag to ease the post processing

After the completion of this request we perform a post processing on the result and replace the tag of each actor by the best ranked tag among its neighbors.

In order to handle the rejection of a generalization between two tags, we add a filter clause in the second block of the UNION clause (line 5 to 7) to exclude the use of a specified broader tag, e.g. `filter(?tag != <http://ademe.fr/energie>)`.

Notice that the analyzed relationship is parameterized and can be replaced by any type of relation defined in the RDF graph (e.g. `sioc:follows`, `rel:worksWith`, `foaf:member`).

2.3 Modularity of an RDF graph

The triples of an RDF description form a directed labelled graph that can be seen as the labelled arcs of an Entity-Relation graph [1], defined as follow:

Definition of an ERGraph: An ERGraph relative to a set of labels *L* is a 4-tuple $G=(E_G, R_G, n_G, l_G)$ where :

- E_G and R_G are two disjoint finite sets respectively, of nodes and relations.
- $n_G : R_G \rightarrow E_G^*$ associates to each relation $r \in R_G$ a couple of entities $e_i, e_j \in E_G$ called the arguments of the relation. If $n_G(r)=(e_1,e_2)$ we note $n_G^i(r)=e_i$ the i^{th} argument of r.
- $l_G : E_G \cup R_G \rightarrow L$ is a labelling function of entities and relations.

Thus, we define the modularity of an Entity-Relation graph as follow:

Definition 2, modularity of an ERGraph: the modularity of an Entity-Relation graph $G = (E_G, R_G, n_G, l_G)$ relative to a set of label *L*, for a given label of relation $p \in L$, is:

$$Q(G,p) = \frac{1}{\left|R_G^p\right|} \sum_{i,j \in E_G} [A_{ij}^p - \frac{d_{<p,i>}^{out}(G) d_{<p,j>}^{in}(G)}{\left|R_G^p\right|}] \delta(c_i, c_j)$$

Where:

- $R_G^p = \{r \in R_G ; l_G(r) = p\}$
- $A_{i,j}^p = 1$ if $\exists r \in R_G^p ; n_G^1(r) = i$ and $n_G^2(r) = j$, 0 otherwise .
- $d_{<p,i>}^{in}(G)$ and $d_{<p,i>}^{out}(G)$ are respectively the number of relations $r^{in}, r^{out} \in R_G^p ; n_G^2(r^{in}) = i$ and $n_G^1(r^{out}) = i$ namely the in and out degree of *i* for the relation labelled with *p*.

We implement this definition of the modularity by querying the RDF graph with SPARQL queries that compute different parts of this formula. In [5], we defined queries to retrieve different network metrics that enable us to compute R_G^p, $d_{<p,i>}^{in}(G)$ and $d_{<p,i>}^{out}(G)$. First we compute R_G^p with a query that simply retrieves the number of pairs of RDF resources that are linked by the property *p*. Then we retrieve the in and out degrees of all the RDF resources linked by a property *p*, with two queries that compute $d_{<p,i>}^{in}(G)$ and $d_{<p,i>}^{out}(G)$ for every possible value of *i*. Finally, we compute the formula by iterating on the results of the two queries below.

The following query retrieves all pairs of connected resources belonging to the same community for the property given as a parameter:

```
1. select ?user1 ?user2 ?tag where{
2.   ?user1 param[property] ?user2
3.   ?user1 scot:hasTag ?tag
4.   ?user2 scot:hasTag ?tag
5. }group by ?user1 ?user2 ?tag
```

The following query retrieves all pairs of disconnected resources belonging to the same community for the property given as a parameter:

```
1. select ?user1 ?user2 ?tag where{
2.   ?user1 scot:hasTag ?tag
3.   ?user2 scot:hasTag ?tag
4.   filter(?user1 != ?user2)
5.   filter(not exists{?user1
param[property] ?user2})
6. } group by ?user1 ?user2 ?tag
```

We then perform a post processing on the outputs of the above queries to compute the modularity of the corresponding community partition.

3. Experiments and Results

In order to validate the benefits of our approach, we applied our algorithm on a dataset of the Ph.D. theses funded by the ADEME. Ph.D. theses have been classified using tags and involve several actors that form a social network made of ADEME employees and academic researchers that collaborate on the funded theses. Academic agents are the Ph.D. students, the Ph.D. supervisors, and the laboratories and institutes they belong to. On the ADEME side, each thesis is followed by an engineer and attached to an internal organization called a "secteur" (sector). Free labels are used to tag the theses, for classifying purposes. From this dataset, we extracted an RDF graph (that comprises both the folksonomy and a description of the network), then we applied our algorithm in order to understand the community structure and activities of the different actors, labeled with the tags that have been used.

3.1 Dataset

The ADEME dataset we analyzed was provided as a relational database and we used the method presented in [5] to build the corresponding RDF descriptions. Figure 3 shows a schema of the concepts we used to represent the ADEME Ph.D. network with the ontologies described in [4] and an ADEME domain ontology that we designed for this analysis. Persons (engineers, students and supervisors) are declared as instances of `foaf:Person` and laboratory and sectors as instances of `foaf:Organization`. The membership of a person to an organization is described with the property `foaf:memberOf`. A student is linked to its supervisor by the property `rel:mentorOf` and to its thesis by the property `dc:creator`. We created the property `ademe:follows`, to link an ADEME engineer to a Ph.D. thesis he follows. Finally, we generated a URI for each tag used to describe a Ph.D. thesis and we used the `scot:hasTag` property to link a thesis to its tags.

Figure 3 describes how we enriched the RDF descriptions of the ADEME Ph.D. theses in order to reveal and structure the corresponding social network. We linked two persons working on the same Ph.D. with the property `rel:worksWith`. We specifically defined the property `ademe:collaboratesWith` to link two agents (`foaf:Person` or `foaf:Organization`) implicated in the same thesis. Two engineers of the same sector are linked with a `rel:colleagueOf` property. We structured these social links by declaring the property `rel:worksWith` as a subproperty of `ademe:collaboratesWith`. Finally, we attached the tags of a Ph.D. to all its involved actors with the property `scot:hasTag`, producing a folksonomy with agents associating tags to thesis. We semantically enrich this folksonomy with the `skos:narrower` relations computed by F. Limpens, on this dataset (the method is outlined in 2.1, a detailed description is available in F. Limpens' Ph.D. thesis [7]).

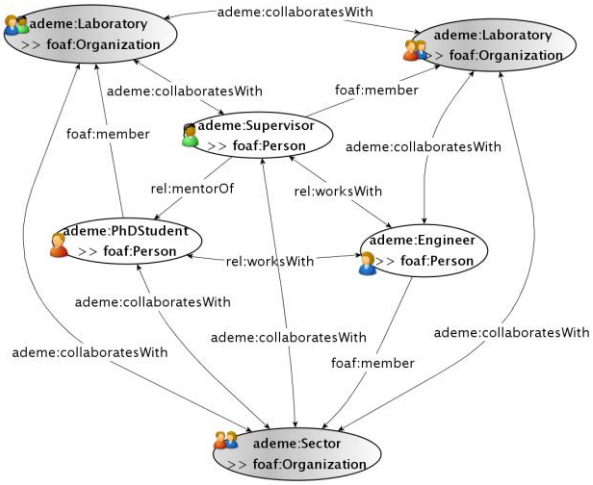

Figure 3. Network from ADEME Ph.D. fundings

3.2 Experiment

We focused our experiment on the sub network of relationships among Ph.D. academic supervisors and ADEME engineers, which are the most active actors of this network. Using the semantic social network analysis method we detailed [5], we measured the characteristics of this dataset:

- 1,853 agents with 1,597 academic supervisors and 256 ADEME engineers.
- 13,982 relationships with 10,246 `rel:worksWith` relations between ADEME engineers and academic supervisors, and 3,736 `rel:colleagueOf` relations between ADEME engineers.
- 6,583 tags, with 3,570 `skos:narrower` relations between 2,785 tags (forming a tree with a depth of 3).

This network is a connected graph that has a diameter of 8, a low density (0,004) and a low clustering coefficient (0,031). This network is highly centralized around the 256 engineers that have a total of 8859 relationships while the 1,597 academic actors have a total of only 5,123 relationships. Indeed, engineers follow several Ph.D. theses and have colleagues inside the ADEME while the most active academic actors supervised a maximum of 14 Ph.D.

In order to evaluate the benefits of introducing semantics in the label propagation, we compared the community that we detected with 4 different algorithms

on this dataset (algorithm 2, 3, 4 are variants we developed for comparison purposes):

1. RAK: random label propagation.
2. TagP (Tag Propagation): propagation of tags without exploiting semantic relations between tags.
3. SemTagP without manual intervention.
4. Controlled SemTagP, which introduces a manual control to avoid the use of some relations between tags. We use the notation SemTagP(tag1, tag2, ...) to specify the tags which `skos:narrower` relations are ignored; e.g., SemTagP(env, energ, model) excludes `skos:narrower` relations with the tags *environnement*, *energetique* and *modelisation*.

We analyzed the evolutions of the modularity of the community partition given by the 4 algorithms and we compared these evolutions in order to observe the added-value of propagating tags (instead of random labels) and exploiting their semantics. Figure 4 presents the curves of the evolution of the modularity of the community partition obtained after each propagation loop. We observe that SemTagP(env, energ, model) offers a community partition, which modularity outperforms the result of RAK, TagP and SemTagP. The RAK algorithm offers the weakest community partition quality on this dataset that is highly centralized with a low density of links. In other words the social links of this datasets are not sufficient enough for revealing the community structure of this social network, using RAK random label propagation. TagP and SemTagP produce community partitions with a significantly better modularity than RAK, however, when considering semantics between tags with SemTagP, we still have a modularity value close to the modularity obtained with TagP. This is due to a very broad tag: *environnement* (environment), that has many `skos:narrower` relations and that aggregates most of the actors in a single community. With SemTagP(env), we exclude the exploitation of `skos:narrower` relations with the tag *environnement*, this considerably improves the modularity value, but with lots of actors in one community tagged with *energetique* (energetic). Finally we obtain a better modularity, 0.12, with SemTagP(env, energ, model) that excludes the use of `skos:narrower` relations of the tags: *environnement* (environment), *energetique* (energetic) and *modelisation* (modeling).

We observe 4 different patterns of tag propagation in the ADEME network that highlight the exploitation of both the link structure and of the emerging semantics of folksonomies. On one side the tag propagation helps partitioning the network into densely linked groups of actors, and on the other side the use of semantic relations between tags helps preserving the identity of small communities, aimed to disappear during the propagation, by gathering them into broader but semantically related communities:

- Most tags used by scattered users in the social network disappear in the first iteration, even if they are used by a large number of users, and do not label a community in the final partition.
- Some tags used by well connected group of users are strengthened by the propagation and still labelling a community in the resulting partition.
- Some tags used by well connected group of users are generalized to broader tags that include and label their community in the resulting community partition.
- Some tags are strengthened by the exploitation of the semantic relations that enable the algorithm to connect semantically related tags and to gather actors working on similar topics but using narrower tags representing different sub topics.

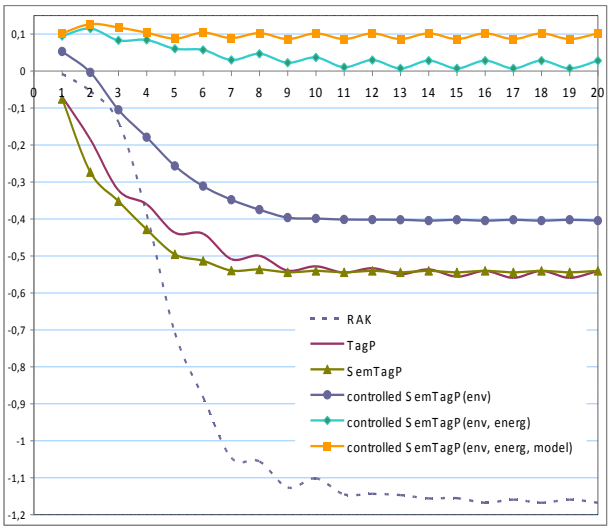

Figure 4. Modularity (Y axis) of the community partition obtained, after each propagation loop (X axis), with RAK, TagP, SemTagP, and 3 controlled SemTagP.

Table 1 compares the size of the communities labelled with 7 tags, initially used by a similar number of users (ranged between 48 and 54), with TagP and SemTagP(env, energ, model). We observe the 4 different propagation patterns described above:

- The tags *évaluation* (evaluation), *photovoltaïque* (photovoltaic) and *innovation* disappeared in both cases because these tags and their `skos:narrower` tags were used by scattered users in the networks.
- The tags *adsorption* and *recyclage* have respectively only 1 and 2 `skos:narrower` relations (with tags used by less than 5 actors).

These tag have not been absorbed during the propagation phase, nor with TagP, nor with SemTagP(env, energe, model).

- The tag *transport* disappeared with both propagations but has been generalized by SemTagP(env, energ, model) to a spelling variant, considered as a broader tag: *transports*, which has 38 `skos:narrower` tags.
- The tag *metaux* (metals) that nearly disappeared with TagP is reinforced with SemTagP by its semantic relations. In particular, this tag has a `skos:narrower` relation with the tag *metaux lourds* (heavy metal) that is used by 75 actors in the initial folksonomy.

Table 1. Comparison of the size of communities labelled with 7 tags (used by a similar number of actors in the initial folksonomy) with TagP and SemTagP (env, energ, model)

Tag	Initial folkso-nomy	TagP	SemTagP(env, energ, model)
adsorption	54	58	15 1 non relevant `skos:narrower` relations with *absorption spectroscopy*.
Evaluation (evaluation)	54	4	0 no `skos:narrower`
Transport	51	1	0 28 `skos:narrower` tags and *transports* `skos:narrower` *transport*
Métaux (metal)	51	2	87 14 `skos:narrower`
Photovol-taïque (photovol-taic)	49	5	0 2 `skos:narrower`
Innovation	48	0	0 6 `skos:narrower`
Recyclage (recycling)	48	8	9 2 `skos:narrower`

Figure 5 presents a visualization of the ADEME social network with the tags of the communities output by SemTagP(env, energ, model). We used a graph visualization tool, GEPHI, with a force layout. The size of the nodes is proportional to their degrees, and the size of the tags is proportional to the size of the labeled communities. Groups of densely linked actors are gathered around few tags, which highlight the efficiency of the algorithm at partitioning the network. Moreover, communities that are labeled with tags representing related topics are close in the visualization, which enable us to build thematic area of the network using the labeling of the communities. In Figure 5, communities displayed in framed area are respectively labeled with tags related to: pollution (1), sustainable development (2), energy (3), chemistry (4), air pollution (5), metals (6), biomass (7), wastes (8). For instance, the area 3 contains tags related to energy production and consumption with the tags *energie* (energy), *silicium*, *solaire* (solar), *moteur* (engine), *bâtiment* (building) and *transports*. This observation shows that SemTagP labeled closest communities with related labels.

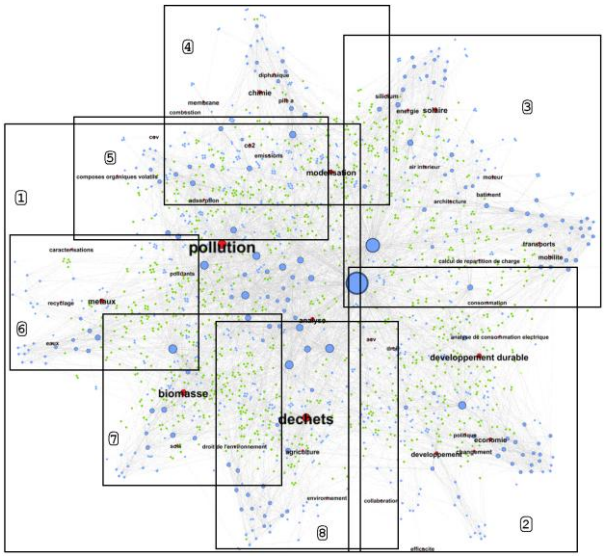

Figure 5. Ph.D. social network of the ADEME with tags labeling the communities obtained with SemTagP(env, energ, model). Red, blue and green nodes are respectively the tags, the ADEME's engineers and the academic supervisors. The framed areas contain communities

4. Discussion

We could go further in exploiting semantic links between tags. (1) In [6] The ADEME's folksonomy was also enriched with `skos:related` and `skos:closeMatch` relations between tags, which exploitation should be investigated. For instance, the triple (*photovoltaic* `skos:related` renewable *energy*), could be exploited to count one more occurrence of the tag renewable energy for each occurrence of the tag photovoltaic. (2) We can exploit other semantic relations between tags and use OWL

entailments such as transitive properties. For instance, SKOS has properties like `skos:transitiveNarrower` (notice that this transitive closure is indirectly performed by the iterative propagation of SemTagP); this could give better grouping of tags but perhaps produce too broad generalizations. Semantic statements like *energy* `skos:transitiveNarrower` *renewable energy* and *renewable energy* `skos:transitiveNarrower` *photovoltaic* could be exploited to count one occurrence of the tag energy for each occurrence of the tag photovoltaic. (3) The ontological primitives used to type the links between actors can describe different intensity of relationships. Consequently when we choose to propagate tags through different properties, we could give more weight to tags propagated through given properties. For instance, in a working environment, tags used by `rel:worksWith` neighbors could be weighted twice more than tags used by `rel:colleagueOf neighbours`. (4) The algorithm may generate disconnected communities labeled with the same tag. This could be a way to detect structural holes [2]. (5) Finally, the current algorithm propagates only one tag per actor, an interesting extension would be to allow several tags to be propagated, which would also allow detect overlapping communities.

5. Conclusion

SemTagP is a novel community detection algorithm that takes benefits of the semantics of RDF descriptions of social networks in order to reveal its communities and to meaningfully label their activities. To our knowledge, this is the first community detection that both detects and controls the labeling of communities. Based on a semantic propagation of tags, SemTagP turns large folksonomies into a subset of significant tags identifying and characterizing communities. The introduction of semantics in the RAK label propagation algorithm offered to handle not only the link structure of social graphs but also the semantics of the tags used by its actors. The label propagation mechanism was designed to exploit the social network link structure and trap labels in dense group of nodes. The assignation of tags, instead of random labels, improves the propagation with the shared vocabulary used to annotate the resources of the network. The exploitation of semantic relations between tags improves the propagation and its control.

We experimented this algorithm on the social network emerging from the Ph.D. theses funded by the ADEME agency, which enabled us to detect and characterize the distribution of its agents and activities. We compared the quality of the partition obtained with 4 different types of propagations: RAK, TagP, SemTagP and a controlled SemTagP. The controlled SemTagP outperformed the results of the 3 others algorithms, highlighting that the introduction of both the tags and the semantics between tags offers a significant improvement to the RAK algorithm.

Many tasks can benefit from this identification of communities of interests in information systems, ranging for instance, from human resources management to notifications and requests routing.

6. References

[1] Baget, J.-F., Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., Gandon, F., Giboin, A.,Gutierrez, A., Leclère, M., Mugnier, M.-L., Thomopoulos, R.: Griwes: Generic Modeland Preliminary Specifications for a Graph-Based Knowledge Representation Toolkit. In:Eklund, P., Haemmerlé, O. (eds.) ICCS 2008. LNCS (LNAI), vol. 5113, pp. 297–310.Springer, Heidelberg (2008)

[2] Burt, RS: Structural Holes. Cambridge University Press, New York (1992)

[3] Corby, O., Faron-Zucker, C.: The KGRAM Abstract Machine for Knowledge Graph Querying. IEEE/WIC/ACM Int. Conference, September 2010, Toronto, Canada (2010)

[4] Erétéo, G., Buffa, M., Gandon, F., Grohan, P., Leitzelman, M., Sander, P.: A State of the Art on Social Network Analysis and its Applications on a Semantic Web. SDoW2008, Workshop at ISWC2008, Karlsruhe, Germany (2008)

[5] Erétéo, G., Buffa, M., Gandon,, F., Corby, O.: Analysis of a Real Online Social Network Using Semantic Web Frameworks, In Proc. Of ISWC'2009,Washington, USA (2009)

[6] Java, A., Joshi, A., Finin, T.: Detecting Communities via Simultaneous Clustering of Graphs and Folksonomies. WebKDD 2008. (2008)

[7] Limpens, F., Gandon, F., Buffa, M.: Helping online communities to semantically enrich folksonomies. In Proc. of WebSci10, Raleigh, USA (2010)

[8] Mika, P.: Ontologies are us: A unified model of social networks and semantics, in Proc. of ISWC'2005, Galway, Ireland (2005)

[9] Newman, M. E. J.: Fast algorithm for detecting community in networks. Phys. Rev. E 69, 066133 (2004)

[10] Leicht, E. A., Newman, M. E. J.: Community structure in directed networks, Phys. Rev. Lett. 100, 118703 (2008)

[11] Raghavan, R.N., Albert, R., Kumara, S.: Near Linear Time Algorithm to Detect Community Structures in Large Scale Network. Phys. Rev. E, 76, 036106 (2007)