Trade-off between complexity of structured tagging and effectiveness of Web service retrieval

Maciej Gawinecki[1], Giacomo Cabri[1], Maria Ganzha[2,3], and Marcin Paprzycki[2]

[1] Department of Information Engineering,
University of Modena and Reggio Emilia, Italy,
name.surname@unimore.it,
[2] Systems Research Institute, Polish Academy of Sciences, Poland,
name.surname@ibspan.pl
[3] Institute of Informatics, University of Gdańsk, Poland

Abstract. Searching for services often starts from the exploration of the service space. Community generated tags can support such exploration. Researchers attracted by the community-available "free manpower" proposed more complex *tagging models*. Those models tag specific parts of the Web service definition: single operations, their inputs and outputs. However, there is no evidence whether the annotation effort is justified by the performance improvement of retrieval (based on those annotations). In this paper we apply similarity-based search to estimate the trade-off between retrieval effort and the annotation effort. Our experiments shows that exploiting similarity metrics for service ranking can compensate missing information and thus be more realistic solution than passing burden of tagging about more aspects on the community.

1 Introduction

Software developers are always short of time and thus need to find a relevant service quickly. A catalog in which services are described accurately, completely and in a concise way by some kind of metadata, may save developers' retrieval effort significantly. No longer developers need to read complex documentation and to test each service candidate. However, building such a catalog requires service brokers to spend their limited funds on providing metadata for each service. It is then important for a service broker to choose such type of metadata that offers good trade-off between retrieval effectiveness and effort, they require to be defined.

Collaborative tagging provides a good trade-off because it moves the burden of describing service behavior to the community [1]. More complex tagging models encourage community to provide metadata also on parts of the Web service definition: single operations, and their input and output [2,3,4,5,6]. Adding more aspects to service metadata allows to find a service not only based on what it does, but also what information it consumes and what information it returns. Adding a structure allows to distinguish between those aspects explicitly during retrieval. The increasing complexity of the tagging models affects the design of user interfaces for tagging as shown in Figure 1. Unlike in traditional models,

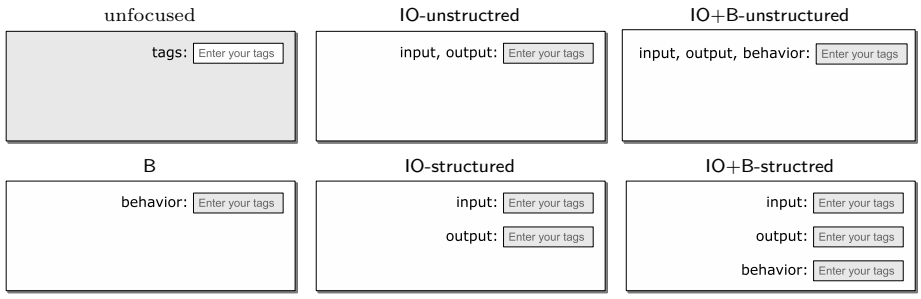

Fig. 1. User interface mock-ups for tagging models: traditional one (unfocused, not evaluated)—open to any aspect of a service—and 5 evaluated tagging models suggesting aspects to describe (Behavior, Input and Output) with and without separation.

a user during tagging is encouraged to focus on selected aspects of a service. Depending on the model she is asked to describe one or more of them in the same or separated fields. Therefore, the user may ask whether effort spent on annotating complex structures is justified by the performance improvement. If not then the community may loose motivation to contribute to the catalog. However, few works on service retrieval using (structured) tags evaluate proposed approaches [3,5], and none estimates the trade-off between the increased tagging complexity, and the retrieval effectiveness.

In this paper we estimate such trade-off to suggest a service broker which tagging model to choose when building the catalog of services. Particularly, we evaluate *whether tagging only about service behavior saves retrieval effort substantially enough or it is worth to encourage the community to describe services with other aspects?* And *whether these aspects should be separated by the user during tagging?* We limit our analysis to single-operation services. We estimate retrieval effort using similarity-based search that operates on service metadata in a uniform way across different tagging models. We estimate annotation effort for each model based on the number of aspects it describes, number of annotations and whether they are structured or not. By estimating those efforts on the corpus of 50 real services from geographic-domain annotated by 27 users we provide suggestions for choosing the right model. Main contributions of this paper are: (1) methodology for estimating annotation effort and retrieval effort for different tagging models; (2) sugesstions for selecting the right model depending on the nature of underlying collection of services.

The paper is organized as follows. Section 2 discusses benefits and limitations of tagging for services; and also user incentives to provide tags. In Section 3 we identify five tagging models from the literature. In Section 4 and 5 we introduce measures for estimating amount of annotation effort and retrieval effort, respectively. Those measures are used in Section 6 to experimentally find a trade-off between both types of effort. Based on our evaluation we provide suggestions for selecting a tagging model in Section 7.

2 Service Tagging Background

2.1 Benefits and Limitations of Tagging for Service Retrieval

In service retrieval the community tags play a number of roles: (i) they provide a network of links to browse services [1], (ii) they provide metadata for ranking-based search [4,3,5] and (iii) they facilitate quick understanding about a service, without reading a complex documentation. Application of collaborative tagging to services has been originally proposed by [1] as an alternative to authoritatively defined taxonomies that are tedious to browse and understand. The advantage of tags is they can capture aspects of a service that are important for the community but has not been encoded neither in the WSDL definitions from service providers, nor in the authoritative classification; moreover, they use vocabulary more relevant and intuitive to a developer [2]. Therefore, the distance between what the developer wants and how it is described in the repository can be minimized [7]. However, tags cannot be used alone for effective retrieval, but only as a complementary mechanism to traditional classification schemes like taxonomies and controlled vocabulary [7]. This goes along with observation of application of tags in other domains, like book search (e.g. Amazon). The possible answer to that problem, being of our concern in this paper, is to *focus* user tagging on particular aspects of a service. [3] proposed to tag input and output of a service separately to facilitate search based on interface matchmaking. In our previous work [5] we proposed to annotate also behavior of services, to match services that are functionally equivalent, but have incompatible interfaces. This approach has provided retrieval performance competitive to solutions based on Semantic Web Services (SWS), presumably thanks to fine-grained aspects of a service, that are easier to express by free-form tags than in formal languages for SWS.

2.2 User Motivations for Service Tagging

Since collaborative tagging relies on voluntary participation, a service broker has no control on which service will be annotated, how and when. Hence, important reasons must attract a critical mass of people to annotate services. In general, reasons to contribute in online communities differ from one community to another. In the following we focus on tagging incentives for communities of two real portals for Web service discovery and tagging. A reader interested in tagging motivations in general and incentives to contribute to collaborative online projects (like open source software or Wikipedia) is referred to works of [8,9,10].

For SeekDa.com Web service search engine the main target group consists of professional software developers who look for web services when they are in a hurry to finish a project and do not have time to develop their own application or address open source solutions [11]. They usually contribute with content (comments, ratings, tags, additional descriptions) during search, mainly to services they found useful in their applications. Tags are provided only if existing ones have been found useful for service retrieval. In this way a user organizes services she found useful; she does not mind to share them with others.

ProgrammableWeb.com online catalog addresses needs of end-user programmers building mash-ups. Searching for a service is the most time-consuming part of application development in their case. It plays also inspirational role steering their projects in particular direction, because from existing mash-ups they can learn what applications can be built and from what services [12]. A user can tag only a service she submits to the community. Hence, tagging is a way to inform the community about own contribution.

Both mentioned portals do not limit aspects on which user may tag a service. Suggesting aspects, and particularly in clearly structured form, provides guidance to a tagger but it also limits types of tags she may enter [13]. It is an open question how such modification of tagging process affects users motivation to tag services.

3 Tagging Models for Web Services

Selecting the right tagging model boils down to choosing a user interface for tagging and deciding how annotations are stored. The interface encourages a user to tag on suggested aspects. It also defines whether the tags on different aspects can be entered into separate fields or into a single one. Annotations from taggers in traditional unfocused tagging model are stored within a folksonomy of services [1]:

Definition 1 *A folksonomy* $\mathbf{F} \subseteq \mathcal{A} \times \mathcal{T} \times \mathcal{S}$ *is a hypergraph* $\mathbf{G}(\mathbf{F}) = (\mathbf{V}, \mathbf{E})$ *with*

- *vertices* $\mathbf{V} = \mathcal{A} \cup \mathcal{T} \cup \mathcal{S}$*, where* $\mathcal{A}$ *is the set of actors (users and the system),* $\mathcal{T}$ *— the set of tags and* $\mathcal{S}$*—the set of services.*
- *hyperedges* $\mathbf{E} = \{(\mathbf{a}, \mathbf{t}, \mathbf{s}) | (\mathbf{a}, \mathbf{t}, \mathbf{s}) \in \mathbf{F}\}$ *connecting an actor* $\mathbf{a}$ *who tagged a service* $\mathbf{s}$ *with the tag* $\mathbf{t}$*.*

Here, a single hyperedge is called an annotation. A single folksonomy may capture different aspects of a service, but does not allow to distinguish between them. To address this limitation, tags entered in different fields are put into separate folksonomies. Therefore, tagging models identified in the literature differ in the user interface (Figure 1) and number of folksonomies to store data:

- **Only the behavior aspect** (B): behavior of a service is described, as originally foreseen in [1]; only a single folksonomy F_b is necessary to model it;
- **Interface unstructured** (IO-unstructured): only input/output are described; only a single folksonomy F_{io} is necessary to model it;
- **Interface structured** (IO-structured): as previous, but aspects are explicitly structured, as defined in [3]; two folksonomies F_i and F_o for input and output, respectively, are modeled;
- **All aspects without structure** (B+IO-unstructured): community describes behavior and input/output, but does not structure this information. Hence, all aspects can be stored in a single folksonomy F_{b+io}.

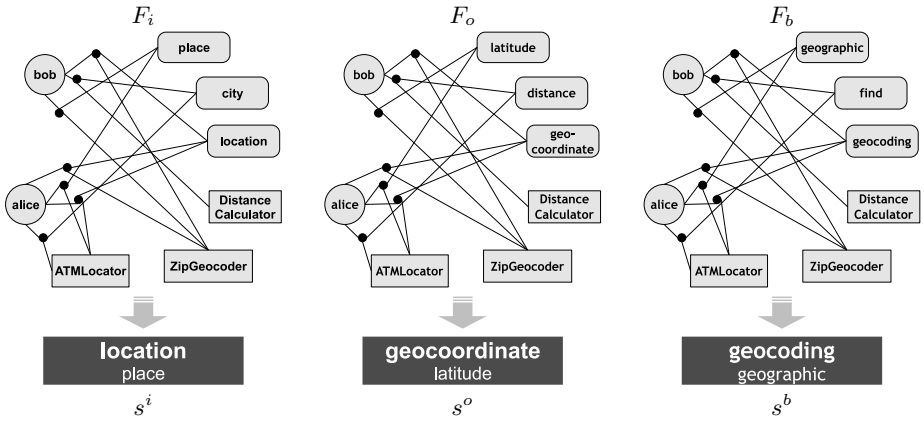

Fig. 2. Example folksonomies (F_i,F_o,F_b) for three service aspects in the B+IO-structured annotation model and corresponding descriptions (tag clouds) of the ZipGeocoder service extracted from them (s^i,s^o,s^b). A circle depicts a tagging user, rectangle—a tagged service and rounded box—a tag used. Each black spot defines a single annotation: which *user* annotated what *service* with what *tag*. A tag cloud is a visual depiction of user-generated tags for a given service part. The bigger the tag, the more users provided it.

– **All aspects structured** (B+IO-structured): as previous, but aspects are explicitly structured to handle ambiguity between different types of tags (as in [5]); hence, three folksonomies F_i, F_o, F_b, are used for input, output and behavior of a service, respectively.

Figure 2 shows examples of folksonomies for B+IO-structured, the most complex model. For instance, F_b is defined by the following annotations: (bob, `geographic`, ZipGeocoder), (bob, `geocoding`, ZipGeocoder), (bob, `find`, DistanceCalculator), (alice, `find`, ATMLocator), (alice, `location`, ATMLocator), (alice, `geographic`, ATMLocator), (alice, `geocoding`, ZipGeocoder).

Obviously, the last model offers the most complex structure and covers all aspects of a service. The remaining models can be created by removing folksonomies that capture unused aspects and/or combining folksonomies to a single one. Such reduction always results in information loss.

4 Estimating Annotation Effort

To understand required user involvement for the given tagging model it is important to understand how much effort it requires from the community to annotate the whole collection of services in the catalog. The annotation process for a single user means: (1) understanding what a service is about based on the provided information (documentation, interaction, monitored QoS, existing applications, other's people tags and comments), and (2) suggesting a combination

of tags describing the service. Quantifying such effort is difficult because of many subjective aspects involved. Particularly, it varies heavily with the background knowledge and cognitive abilities of users, their familiarity with annotation tools, usability of such tools and of available information. This makes it difficult to capture the cost in a general way. Centralized annotation offers two measures that allow to capture objective aspects of the annotation effort: *absolute*, defined as a number of man-hours spent on a complete annotation of a whole collection of objects [14] and *relative*, defined as a fraction of possible annotations to collect [15]. Man-hours are not a good measure of effort of collaborative tagging, as it is almost impossible to measure them in the open (i.e. uncontrolled) distributed environment, with ad-hoc voluntary participants. The relative measure will not work because there is no criterion of completeness, as tags are added continuously [16]. Therefore, we propose to estimate annotation effort for each model based on: the number of aspects it describes, number of annotations its instance contains and whether they are structured or not. We assume the annotation required more effort from the community, if more aspects have been captured and more annotations have been provided. For simplicity, we assume that this is the same amount of effort for a human to provide 10 annotations as for ten people to provide a single annotation each.

5 Estimating Retrieval Effort

In ranking-based retrieval the less effective a solution is, the more results in the ranking a searcher needs to scan to find a perfect one. In other words, she needs to spend more retrieval effort. The normalized Discount Cumulative Gain (nDCG) [17] curve models retrieval effectiveness and allows to estimate amount of retrieval effort. Let us define first G_i, a gain value that the i-th returned service obtains for being relevant. We define

$$DCG_i = \begin{cases} G_1, & i = 1 \\ DCG_{i-1} + G_i/log_2(i+1), & i \geq 2 \end{cases} \quad (1)$$

The Discount Cumulative Gain (DCG) rewards relevant answers in the top of the ranking more than those in the bottom of the ranking. Calculated DCG_i is then normalized by the ideal possible DCG_i to make the comparison between different approaches possible. When plotting the $nDCG$ curve, the X-axis represents a rank, and the Y-axis represents a $nDCG$ value for a given rank. An ideal similarity approach has a horizontal curve with a high $nDCG$ value. Hence, the size of the area above the actual $nDCG$ curve tells how much more effort a user needs to spend on retrieval with respect to scanning the ideal ranking. The discount factor of $log_2(i+1)$ is relatively high, to model an impatient user who gets bored when she cannot find a relevant answer in the top of the ranking.

Retrieval effectiveness depends on the following variables: a user query formulation strategy, a ranking algorithm, an tagging model and its content. Since we are interested in attributing estimation of retrieval effort to the last two variables, we need to eliminate impact of the first two.

5.1 Eliminating Impact of Query Formulation Strategy

Users differs in how they formulate the same information need with query keywords and thus the choice of a user may impact retrieval effectiveness. Similarity-based search does not has this drawback, because a request is expressed as a service, for which services of similar functionality have to be returned. Similarity-based search ranks results with respect to they similarity to the request. Existing methods in the literature operates on different metadata describing a service: WSDL definition (e.g. [18]), keywords extracted from WSDL definitions (e.g. [19]), or semantic annotations (e.g. [20]). In our case metadata are defined by community annotations.

We define similarity measure for the tagging model, where all annotation aspects are explicitly structured (B+IO-structured). By s^i, s^o, s^b we denote aspects of the tagged service s, concerning input, output and behavior, respectively. They can be extracted automatically from the corresponding folksonomies F_i, F_o, F_b. For instance, Figure 2 shows descriptions extracted for the ZipGeocoder service. By λ we denote similarity metric for the tagged descriptions from a single folksonomy. Since different aspects are described in different folksonomies, we can measure similarities for each aspect of two services s_1 and s_2 separately and then combine similarities together:

$$\gamma^{b+io-structured}(s_1, s_2) = \lambda(s_1^i, s_2^i) + \lambda(s_1^o, s_2^o) + \lambda(s_1^b, s_2^b) \tag{2}$$

For unstructured annotations models, operating on a single folksonomy, we reduce similarity measure between s_1 and s_2 to $\lambda(s_1, s_2)$. For the IO-structured tagging model, using two folksonomies, we define it as: $\gamma^{io-structured}(s_1, s_2) = \lambda(s_1^i, s_2^i) + \lambda(s_1^o, s_2^o)$.

5.2 Limiting Impact of Ranking Algorithm

We want to define the similarity measure λ for any two resources (based on the information about them) in a *single* folksonomy. Depending on whether the tagging model is structured or not, a *resource* can be a single service (unstructured models) or a single aspect of a service (structured models). To avoid a bias towards a particular ranking algorithm we selected a number of similarity measures λ proposed by Markines et al. [21] for tagged resources. We selected two that provide relatively good accuracy, while are still simple to analyze: the projection-aggregated overlap similarity, and the distribution-aggregated cosine similarity. These measures differ in how they aggregate information about a resource from a folksonomy hypergraph, and how they compute similarity based on the aggregated information. *Projection-aggregated overlap similarity* considers two resources to be similar, if they share many tags. *Distribution-aggregated cosine similarity* considers not only the overlap in tags, but also how they are distributed among users and other services. There are three variants of cosine similarity. A *term frequency* (TF) variant considers two resources similar, if tags they share have been proposed by many users. An *inverse document frequency*

(IDF) variant considers two resources similar, if the tags they share are uncommon in the collection. A *combination* of those two (TF/IDF) considers two resources similar, if the tags they share have been proposed both: (a) for them but not for many other resources and (b) by many users. Only the TF variant has been evaluated in [21].

6 Experimental Evaluation

The goal of the preliminary evaluation was to check which tagging model offers the best trade-off between the annotation effort and the retrieval effort.

6.1 Test Data

Our test data consisted of the collection of services and their annotations. For the first part we used the Jena Geography Dataset (JGD50) [22] containing WSDL definitions of 50 real data-centric single-operation services from the geography domain. It is more representative for a class of domain-specific catalogs of homogeneous services (e.g. GeoNames.org) rather than for class of catalogs with a large number of diverse services (e.g. SeekDa.com). Moreover, for a single user the homogeneity of the collection means that she must provide more tags for a single service to differentiate it better from other similar services. In the simplest case, if each service was from a different functionality domain, then it would be sufficient to tag it with only its domain name.

The data set contains also 9 service requests. We filtered those 7 of them, for which the set of relevant services (with respect to relevance judgments) included a large fraction of services of incompatible interfaces but still with equal or approximate functionality. Since original service requests were expressed in a natural language, and similarity search requires a service as a request, we replaced each original service request in the JGD50 with the most relevant service offer (based on the corresponding relevance judgments). To estimate relevance of results we used graded relevance judgments from the dataset. They fit well for similarity-based search where even partial match in functionality, can be somehow relevant to the user. We used the relevance setting that emphasized functional equivalence over interface compatibility, because it models well users that are interested in services providing similar functionality, even if they are not interface compatible; moreover, such users are not interested in interface compatible services that do not provide similar functionality.

At the time of the evaluation, there was no industrial corpus of structured tags for services placed in the test collection. Note that lack of such corpus is the most challenging obstacle in evaluation of retrieval based on collaborative tagging of Web services [1,2,4,6,3], and software components in general [7]. For the B+IO-structured model we reused the WSColab[4] artificial corpus, which contains 4008 annotations for JGD50 services. It consists of both annotations

[4] `http://fusion.cs.uni-jena.de/professur/jgdeval/JGD50-WSColab-Services.zip`

tagging model	#total	#input	#output	#behaviour
B	1496	0	0	1496
IO-*	2512	1437	1075	0
B+IO-*	4008	1437	1075	1496

Table 1. Numbers of annotations for evaluated instances of tagging models.

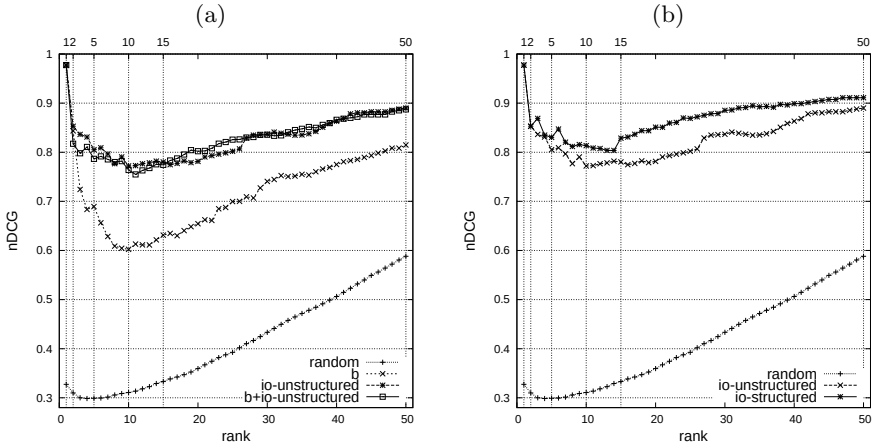

Fig. 3. Effectiveness of overlap similarity when adding more: (a) aspects to unstructured models, (b) structure to IO-based model. Averaged over all queries.

bootstrapped automatically (32%) and provided by the community (68%). The latter were collected in the open (non-laboratory) environment through the collaborative tagging portal from 27 users through the collaborative tagging portal based on WSDL documentation (see, also [23], for more details). For the remaining tagging models the tag corpora were created from the B+IO-structured model as described in Section 3. Numbers of resulting annotations for creating corpora are reported in Table 1.

6.2 Experimental Results

We compared similarity search results for the five tagging models identified in Section 3. Each model was evaluated using both similarity metrics: the overlap, and the cosine similarity (all variants). This resulted in 20 experimental cases. To put evaluation results into the context we plotted a curve for a random similarity search. The curve has been generated by averaging effectiveness over 50 random permutations of ranking. Generating another 50 permutations has shown that this is a sufficiently large base for stable results in this test collection. The curve is relatively high, confirming that services in the test collections are similar to each other and thus challenging to distinguish for taggers and a retrieval algorithm.

Firstly, we analyzed whether adding more aspects limits retrieval effort. Figure 3a shows nDCG curves for unstructured models with a different number of

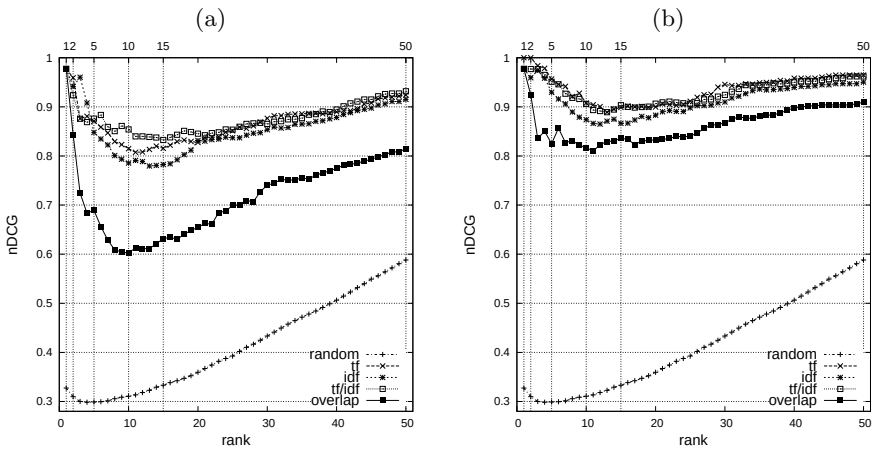

Fig. 4. Effectiveness of varying similarity metrics for: (a) the B model and (b) the B+IO-structured model. Averaged over all queries.

aspects. It demonstrates that, for the simplest similarity metric, IO-unstructured allows to save about 11% more retrieval effort than B, but combining those aspects together into IO+B-unstructured does not significantly save retrieval effort with respect to IO-unstructured: much below 1% (evaluation has shown that structured models behave in an analogous way). This is suprising because the large fraction of services in the corpus is functionally equivalent but not necessary interface compatible and adding behaviour tags should help improve effectiveness. A possible reason is that behaviour tags are more redundant: adding behaviour aspect introduces less than 14 new tags per a service (users often describe output of data-centric services as their behaviour), while adding input and output aspects to B introduces more than 20 new tags.

Secondly, we investigated whether adding structure limits retrieval effort. Adding a structure allows to disambiguate between tags describing different aspects, but may cause errors caused by structure gap. This is because the structure itself can be ambiguous for an annotator (we recall the example where output tags are found among the behavior tags). Figure 3b shows nDCG curves for IO-* models. It demonstrates that adding a structure saves only 4% of retrieval effort for the simplest similarity metric (evaluation has shown that IO+B-* models behave in the same way: about 4% of retrieval effort was saved). A possible reason is that only 12% of tags describing interface between input and output overlapped (on average for a service), which means that remaining tags provided reasonably good disambiguation comparing to the structure.

Thirdly, we investigated whether previous observation can be generalized to different similarity metrics. Across all tagging models, the projection overlap required more retrieval effort than any variation of the cosine similarity. This is consistent with results of Markines et al. [21], though in our case the additional information about tag distribution in the folksonomy appears to have larger impact

on the effectiveness of retrieval. For instance, Figure 4a shows nDCG curves of varying similarity metrics for the B model. The possible interpretation of such a large impact is that the projection overlap, in opposition to cosine similarity, does not distinguish between minor (e.g. `miles`), and major features (e.g. `geocoding`) and thus may incorrectly detect two services as highly similar based only on a minor feature. Major features are presumably those which are represented by tags on which more people have agreed. We have found also that among variants of cosine similarity there is no dominating winner. Moreover, we observed that using cosine similarity for more complex tagging models does not have such significant impact on retrieval effort increase. Comparing changes in retrieval effort between the B model (Figure 4a) and the IO+B-structure model (Figure 4b) demonstrates that fact. The conclusion is that selection of the right similarity metric can compensate lack of some aspects and structure in the tagging model.

Fourthly, having selected competitive similarity metric (TF/IDF cosine similarity), we related retrieval effort to the number of annotations (see Table 1) for the worst (B) and the most effective (IO-B-structured) model. Surprisingly, even if IO-B-structured allowed to take only 6% retrieval effort less than B, it still required significantly more annotation effort: it used about 2.7 times more annotations and three service aspects instead of one.

7 Conclusions and Suggestions for Service Brokers

Choosing the right tagging model is important for a service broker relying on the participation of the community. The performed experiments show that, for the considered corpus of service, the tagging model describing only behavior offered the best trade-off between retrieval effort and annotation effort. The amount of retrieval effort saved by tagging on more aspects is not impressing, especially when related to the annotation effort. Therefore, for *similarity search* in general, it makes sense to add more aspects to the tagging interface incrementally, only if existing ones do not allow to match or disambiguate services in the service catalog. Furthermore, exploiting similarity metrics for service ranking can compensate missing information and thus be more realistic solution than passing burden of tagging about more aspects on the community. Finally, using structure for retrieval did not limited retrieval effort substantially as other tags could play similar disambiguation role. However, the structure can be still for important for searches focused on selected aspects or for a tagger who needs clear separation between aspects during tagging.

Acknowledgments We thank to the members of FUSION group from Jena University for discussion and comments that helped strengthen the paper.

References

1. Meyer, H., Weske, M.: Light-Weight Semantic Service Annotations through Tagging. In: ICSOC. (2006) 465–470

2. Fernández, A., Hayes, C., Loutas, N., Peristeras, V., Polleres, A., Tarabanis, K.A.: Closing the Service Discovery Gap by Collaborative Tagging and Clustering Techniques. In: SMRR. (2008)
3. Bouillet, E., Feblowitz, M., Feng, H., Liu, Z., Ranganathan, A., Riabov, A.: A Folksonomy-Based Model of Web Services for Discovery and Automatic Composition. In: IEEE SCC. (2008) 389–396
4. Chukmol, U., Benharkat, A.N., Amghar, Y.: Enhancing Web Service Discovery by Using Collaborative Tagging System. In: NWESP. (2008) 54–59
5. Gawinecki, M., Cabri, G., Paprzycki, M., Ganzha, M.: WSCOLAB: Structured Collaborative Tagging for Web Service Matchmaking. In: WEBIST. (2010)
6. Silva-Lepe, I., Subramanian, R., Rouvellou, I., Mikalsen, T., Diament, J., Iyengar, A.: SOAlive Service Catalog: A Simplified Approach to Describing, Discovering and Composing Situational Enterprise Services. In: ICSOC. (2008) 422–437
7. Vanderlei, T.A., Durao, F.A., Martins, A.C., Garcia, V.C., Almeida, E.S., de L. Meira, S.R.: A cooperative classification mechanism for search and retrieval software components. In: SAC, ACM (2007) 866–871
8. Kuznetsov, S.: Motivations of contributors to wikipedia. SIGCAS Comput. Soc. **36**(2) (2006) 1
9. Marlow, C., Naaman, M., Boyd, D., Davis, M.: HT06, tagging paper, taxonomy, Flickr, academic article, to read. In: Hypertext. (2006) 31–40
10. Preece, J.: Sociability and usability in online communities: determining and measuring success. Behaviour & Information Technology **20**(5) (2001) 347–356
11. Cuel, R., Oksana, T.: SeekDa Case. Technical report, Insemtives Project (2010)
12. Hartmann, B., Doorley, S., Klemmer, S.R.: Hacking, mashing, gluing: Understanding opportunistic design. IEEE Pervasive Computing **7**(3) (2008) 46–54
13. Bar-Ilan, J., Shoham, S., Idan, A., Miller, Y., Shachak, A.: Structured vs. unstructured tagging a case study. In: Proc. of the Collaborative Web Tagging Workshop (WWW '06). (May 2006)
14. Küngas, P., Dumas, M.: Cost-Effective Semantic Annotation of XML Schemas and Web Service Interfaces. In: IEEE SCC. (2009) 372–379
15. Gomadam, K., Ranabahu, A., Ramaswamy, L., Sheth, A.P., Verma, K.: Mediatability: Estimating the Degree of Human Involvement in XML Schema Mediation. In: ICSC. (2008) 394–401
16. Halpin, H., Robu, V., Shepherd, H.: The complex dynamics of collaborative tagging. In: WWW. (2007) 211–220
17. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. **20**(4) (2002) 422–446
18. Wang, Y., Stroulia, E.: Semantic Structure Matching for Assessing Web-Service Similarity. In: ICSOC. (2003)
19. Dong, X., Halevy, A.Y., Madhavan, J., Nemes, E., Zhang, J.: Similarity Search for Web Services. In: VLDB. (2004)
20. Hau, J., Lee, W., Darlington, J.: A Semantic Similarity Measure for Semantic Web Services. In: Web Service Semantics Workshop at WWW. (2005)
21. Markines, B., Cattuto, C., Menczer, F., Benz, D., Hotho, A., Stumme, G.: Evaluating Similarity Measures for Emergent Semantics of Social Tagging. In: WWW. (2009) 641–641
22. JGD: Jena Geography Dataset. `http://fusion.cs.uni-jena.de/professur/jgd`
23. Gawinecki, M., Cabri, G., Paprzycki, M., Ganzha, M.: Evaluation of Structured Collaborative Tagging for Web Service Matchmaking. `http://mars.ing.unimo.it/gawinecki/pubs/wscolab-manuscript.pdf` (2010)