Breaking free from your information prison

A recommender based on semantically enriched context descriptions

Jonas Lutz, Barbara Thönssen, Hans Friedrich Witschel
University of Applied Sciences and Arts Northwestern Switzerland
FHNW
Olten, Switzerland
{jonas.lutz, barbara.thoenssen, hansfriedrich.witschel}@fhnw.ch

***Abstract*—Information repositories, implemented as Enterprise Portals (EP) on the intranet, are increasingly popular in companies of all sizes. Enterprise Portals allow for structuring information in a way that resembles the organization of paper copies, i.e. simulating folders and registries and furthermore, provide simple routines for publishing and collaborating. Hence, in general, such kind of information management is not much different from paper management: electronic documents must be uploaded into the Enterprise Portal manually, filed into folders (which have to be created manually, too), tagged and related to other information objects if need be. With this approach information structuring remains subject to the individual user leading to the well-known problems of multiple filing, overlooking relevant information and incomprehensible folder structure. The SEEK!sem project aims at improving such kind of information system by automatically identifying and recommending related information resources to be added to a folder. The recommendations are based on rules, exploiting content and context similarity of information resources. Rules can be created upfront, based on explicitly defined relations between information objects. They can also be machine learned, i.e. the recommender exploits the existing linkage between documents, folders and other objects to learn "relatedness rules". In either case, potential new connections are inferred by applying the rules in a reasoning step. Recommended new connections are ranked by the sum of the scores of all applied rules – the rule scores, again, can either be provided by experts or machine-learned. The applied rules can serve as an explanation of a recommendation, i.e. they can assist users in understanding why a particular connection is suggested.**

Keywords—information management, similarity, machine learning, context

I. INTRODUCTION

The SEEK!sem project is a Swiss national funded research project[1]. Business partner in the project is a Swiss software vendor who offers a web-based information management system. Electronic documents (i.e. text but also images) can be uploaded into the Enterprise Portal manually, filed into folders, tagged and related to other information objects if need be. Even though the current information system enables storing and accessing information objects in a well-structured user-friendly way, information management remains subject to the individual user - leading to the well-known problems of incomprehensible folder structure, multiple filing and overlooking relevant information. In the paper at hand we focus on how users can be supported in identifying related information in large repositories. That is, when a user is working with a folder, the system should proactively recommend resources that are related to this folder. Therefore we exploit content and context similarity of information resources and recommend related information objects based on rules.

The context of an information resource is given by its relation to other objects in the system. Since we are considering an Enterprise Portal (or other information management system used within an enterprise context), the majority of object classes and their relations are related to concepts in the business world. We therefore use an enterprise ontology called ArchiMEO (concepts and relations are derived from the ArchiMate standard [1]), which has been adapted to include all classes and relations that are present in the studied information management system.

For explaining to the user why a resource was recommended and how it was scored, the rules are represented in natural language, based on the names of the relations in ArchiMEO. If a user accepts a recommendation her decision will be stored and later used for improving the recommendation rules.

Outline of the paper is as follows: in Section II, we introduce the SEEK!sem project to set the frame for our research. We then outline the problem statement, research questions and research methodology in Section III. Related work is presented in Section IV. In Section 0 we describe our approach for recommending information resources in detail. Section VI discusses the results of a preliminary evaluation. The paper concludes with Section VII.

II. BACKGROUND: THE SEEK!SEM PROJECT

The SEEK!SDM software already has a good market penetration in Switzerland and has been successfully introduced in public administrations, such as the city council of Zurich. However, with today's software version, management of electronic documents is not much different from paper management. Many tasks are to be performed manually and the potential of having machine readable information is not exploited yet. Thus, key points of

[1] Kommision für Technologie und Innovation KTI. Project no 14604.1 PFES-ES.

improvement are (a) increasing the degree of automation in the creation and filing of information objects, e.g. through automatic classification of information objects, (b) improving the transparency of information in its context through search improvement and recommendations, e.g. on the basis of similarities, previous information usages or semantically enriched enterprise context information and (c) continuous, automatic discovery and alignment of (hidden) structures, e.g. through the implementation of unsupervised learning algorithms.

Within the SEEK!sem project these three points are addressed. In the paper at hand we focus on (b), i.e. how information supply and, as a consequence, the structures within the information management system, can be improved by automatically recommending related information objects.

The SEEK!sem project aims at improving the existing information management software in order to strengthen the business partner's market position and increase the user's value creation.

III. Research questions and approach

A. General problem description

Assume that a user has created a folder (which we will call "dossier" later on) and filed a number of new information objects (documents, images, etc.) into that folder. When working with the folder, other, already existing information objects that could be of interest will be recommended. By accepting the suggestion, the recommended and the newly added information objects are automatically linked.

For recommendation similar information objects must be identified. Whereas similarities between text documents are quite easy to implement – mostly based on the overlap of words within the documents – and many approaches are already available, similarities between information objects of mixed format, for example between text documents and images, images and structured information like contact and address data, and information objects and information structure (i.e. text document and folder access rights) are rather hard to determine.

B. Research questions

These challenges result in the following research questions to be addressed within this work:

1. How to describe the context of information objects in a sound and machine understandable way?
2. How to exploit this context to establish reliable similarity between arbitrary information objects and use it for recommending related resources?

C. Research method

In this work, we have designed a framework and algorithm a) to represent information objects and their context (see research question 1 above) and b) to exploit that context for learning a similarity measure between information objects (see research question 2).

We followed the design research principles as proposed by Hevner [2], which means that we traversed the following phases:

- We first analysed the required properties of the similarity measure as well as the available context information that should be represented ("awareness of problem" phase)
- We then put forward and implemented our data representation, similarity measure and recommender algorithm ("suggestion" and "development" phases)
- We finally evaluated the resulting software prototype by applying it to real-world data. The resulting rules (or paths as we will call them later) that were discovered by the algorithm on the given data were inspected manually and discussed with our business partner ("evaluation" phase). Because the proposed similarity measure has not yet been integrated into a operational recommender, we could only test it in this qualitative way.

In this paper, we present the details of our algorithm (including recommendation), as well as the qualitative results that we obtained during the evaluation of the proposed similarity measure.

IV. Related Work

Recommendation of related items, such as of related products in an on-line shop (cf. [3]) can be transferred to the domain of information resources, e.g. to recommend legal documents that are related to a given legal case a lawyer is working on [4],[5]. Such recommendations can ease the work within a short timeframe (as in the case of the lawyers), but it can also help to persistently organize information in a more consistent way. Hence, retrieval of this information becomes easier on a longer term basis and emergence of information silos is avoided.

In any case, the notion of *similarity* between enterprise objects – such as documents and/or folders, but also others such as images or (information about) persons – is at the heart of many recommenders. More generally, similarity also plays a major role in a vast number of tasks in information retrieval, data mining and text mining.

Traditional measures of similarity or distance are formulated as functions that combine the similarity or distance of individual attribute (or feature) values via a weighted or non-weighted sum. For objects that are represented as vectors with numeric features, dot products (as used in information retrieval, e.g. [6]) or Euclidean distance e.g. used for clustering) are generic examples of such functions.

When using graph-based representations of data, a standard approach is to establish similarity between two graph nodes n1 and n2 as the probability of reaching n2 when starting a random walk from n1 (e.g. [7]).

All these approaches have in common that similarity is established on a *sub-symbolic* level, i.e. by adding and multiplying numbers that quantify the strength of certain (symbolic) relations. In this section, we therefore study previous work that has addressed the two main drawbacks of these traditional sub-symbolic approaches, namely

- their inability to *explain* in an intuitive way why two objects are similar
- and, in cases of functions relying on a weighted sum, their heuristic and often sub-optimal way of deriving weights.

Much of the work that has been done to address the first drawback stems from the area of case-based reasoning (CBR). A CBR system is a software system that supports humans in solving a current problem by retrieving problems (cases) that are similar to the current one and that have been successfully solved in the past. In order for a human to be able to make an informed choice among retrieved cases and to adapt it to her needs, it is particularly important for a CBR system to explain why the retrieved cases were considered similar to the input case.

In [8], an approach is developed that uses a symbolic similarity measure and summarises retrieved cases by all features that they – and the input problem – have in common.

A particular form of symbolic similarity, namely rule-based similarity is introduced by Sebag and Schönauer [9]: they assume a set of rules to be defined over the attributes of cases in CBR. Rules that match case descriptions are used to predict the plan (i.e. a class of solutions) that will be adapted to a case. The similarity of two cases is defined as the sum of weights of all rules that match the problem description of both cases.

The work in [9] additionally emphasises why rule-based similarities are more flexible than weight-based similarities: apart from giving an explanation of similarity, they allow to express the influence of *dependencies* between attribute values on similarity, whereas weight-based similarity assumes that all attributes contribute to similarity independently with a certain strength.

The second drawback of many weight-based similarity functions, namely the heuristic way of establishing weights, can be addressed by approaches that *learn* a similarity function. These approaches exploit instances of known similarity, i.e. pairs of objects that have been manually labeled as being similar. This is used e.g. in image retrieval [10] where pictures representing the same object are manually identified and a metric is learned from such training examples that will help to identify similar pictures automatically.

Using a graph-based representation of data, Minkov and Cohen [11], [12] have experimented with learning similarities in graphs with typed edges. Their approach is based on the random walk paradigm where a walk starts at a given set of initial nodes (the *query*). Transition probabilities depend on the types of edges. After some steps, the resulting distribution of the probability mass over the graph's nodes is interpreted as scores and thus results in a ranking of the graph nodes w.r.t. the query. This has been applied e.g. in the task of finding synonyms in word graphs [13].

As training data, they consider example queries, together with all nodes labeled as relevant to these queries. The training data is used to learn the transition probabilities (i.e. graph weights) in [11]. In a later work [12], this purely sub-symbolic learning is enhanced with a so-called *path tree* which is used to constrain a random walk by preferring paths that are more likely to lead to relevant nodes. *Paths* are – as in our work – defined as sequences of edge types. Each path is assigned a (smoothed) probability of leading to a relevant node.

Although the set of paths that lead to a retrieved node in a random walk could be used as an explanation of why that node was retrieved, Minkov and Cohen do not consider this possibility.

To our knowledge, the only work that combines symbolic similarity with learning is that of Sebag and Schönauer: in [9], their definition of rule-based similarity (see above) is complemented with a machine learning approach that learns the rules from the examples in the case base.

In our work, we present a new form of such combination: the result of our approach can be understood as a rule-based similarity measure that is being learned from manually labeled examples. The approach differs from the one in [9] in that it operates on graphs. It uses a very special form of rules, namely ones that predict new edges between two nodes based on the presence of certain paths (sequences of edge labels) between these nodes. This allows establishing similarity between objects that do not share any immediate attribute. Hence, it is an even more general approach than the one in [9].

The use of ontologies for improving information retrieval has been extensively studied with the spread of the internet since the late 90s. Focus of the research was the (mostly manual) semantic description of web pages to improve search on the Internet (e.g [14] and [15] studied the (semi-automatic) classification of Web pages based on their structure and used the formal semantics of an ontology for consistency checking and filtering of web pages). See [16] for a comprehensive overview and comparison of tools and frameworks for semantic annotations. The use of an enterprise ontology for exploiting organisational context for the automatic creation of metadata was researched by [17], [18] and [19].

An enterprise ontology, called ArchiMEO, was developed within a PhD thesis [20] based on former work of [21] and [19]. ArchiMEO represents the notation of the Enterprise Architecture Standard 'ArchiMate' [1]. ArchiMate offers an enterprise model and a formalized, but not machine understandable language for model description (an adaptation of UML classes). For using an enterprise architecture description on operational level, e.g. providing the context of information objects, this kind of representation is not enough. Therefore the ArchiMate concepts and relations were transferred into an ontological representation and expanded taking into account existing work (e.g. [22], [23], [24], [25] and [26]). The ArchiMEO ontology is used in the SEEK!sem project to provide the meaning of concepts and relations of the "information graph" and thus explain the context of information objects.

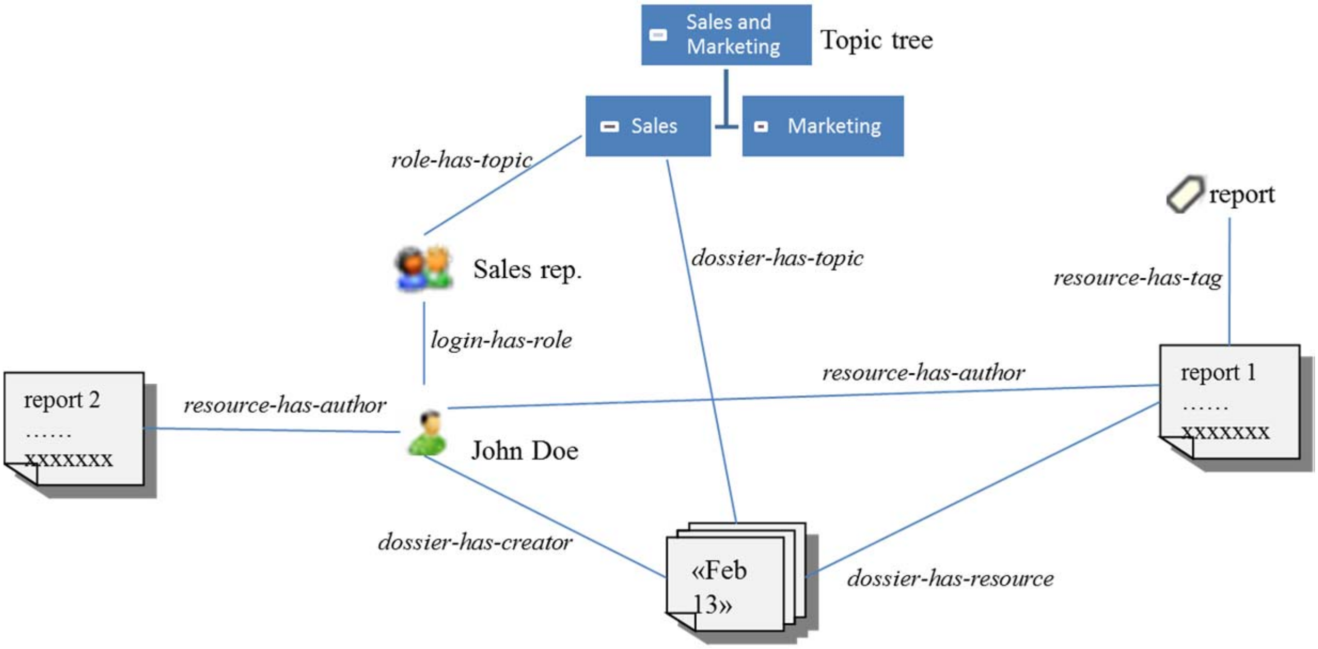

Figure 1. Information resources represented as a graph

V. SEEK!sem Recommender

A. Initial situation

As described above, we consider the situation of an Enterprise Portal – or more generally an information management system – where various types of information objects are organized.

Since SEEK!sem is considered the next generation of the existing information management software SEEK!SDM, our research is based on a test implementation provided by the business partner. In the concrete implementation, there are information objects, so-called *resources*, which need to be organized. Resources can be textual documents, multimedia content, but also information on persons, organisations, events and tasks. On the other hand, the system provides elements for structuring and organizing resources (*structure elements*). The most important structure element is called a *dossier,* which can be understood as a folder that holds resources related to the same task or business process. Other structure elements are topics from a taxonomy used to hierarchically organize dossiers and a flat set of tags that can be assigned to any resource. Finally, users can be assigned to roles, which are used to manage access rights within the system.

Besides directly allocating resources to dossiers, users can establish additional links between resources and/or structure elements by selecting relations from a predefined set, such as authorship, role membership or tagging.

Note that some of these elements and relations can be found elsewhere, e.g. in a file system where folders can be seen as equivalents to dossiers and file attributes (e.g. creator) are considered meta-data.

Figure 1 shows an example dossier (“Feb13”) which has one resource (“report 1”) and belongs to the topic “Sales”. The resource “report 1” is tagged as a “report” and has been authored by John Doe. The dossier has been created by the same John Doe who has the role of a sales representative, which in turn is connected to the topic “Sales”. John Doe has also authored “report 2”, which is, however, not assigned to any dossier.

B. Representation of information objects and context

We now address our first research question, namely the challenge of how to represent information objects and their context in a machine-understandable way.

By looking at Figure 1, it is obvious that the resources and their relations can be represented as a graph with typed nodes and edges. We call this form of graph a *resource graph*. As mentioned in Section IV above, we have mapped the node and edge types to the concepts and relations that are available in the ArchiMEO enterprise ontology. Thus, we will later be able to represent the graph as an RDF graph and apply reasoning procedures.

C. Definitions and general approach

The goal of our research was to support users of an information management system in organizing resources by recommending objects belonging into the same dossier. That is, when a user is working with a given dossier D, the system should proactively recommend resources that are related to D. The recommended resources should be ranked by their (estimated) similarity with D.

We assume that a large number of resource-dossier assignments has been performed manually before our system comes into play. We further assume that these manual assignments represent true similarity, i.e. that they can be used as a training set for learning a measure of similarity between a dossier D and a resource R. We call each pair (D,R) where R is manually assigned to D a *positive example*. All other pairs (D',R') where there is no assignment (yet) are called *negative examples*.

Below, we will propose to learn a similarity measure that is based on the structure of the resource graph. More precisely, the similarity is expressed in the form of *abstract paths*. We define an abstract path as a sequence of edge types. A path p in a concrete resource graph is said to *correspond* to an abstract path *p'* (and vice versa) if the types of p's edges are equal to the edge types in *p'*. We denote this by $p \cong p'$. In the example resource graph in **Figure 1**, the abstract path that corresponds to the shortest path between the tag "report" and John Doe is (*resource-has-tag*, *resource-has-author*). Note that we neglect the direction of edges.

D. Ourline of proposed algorithm

With this, we are able to address our second research question, namely how to exploit context to establish reliable similarity between arbitrary information objects and use them for recommending related resources.

We establish a resource recommender by using the following steps:

1. Using the training data of manual dossier-resource assignments, we identify those abstract paths that are most likely to connect positive examples. Associate to each abstract path p a score s_p that reflects how well the path can distinguish between positive and negative examples.
2. For a given dossier D, we retrieve and rank a set of recommended resources $\{R_1, \ldots, R_n\}$ in the following way:
    a. Using a reasoner, we identify all acyclic paths starting at D that correspond to any of the abstract paths identified in step 1. We collect the endpoints of these paths into the set of candidate resources $\{R_1, \ldots, R_n\}$.
    b. Now, for each candidate resource R_i, we define the similarity between D and R_i based on the scores of abstract paths corresponding to the paths that connect D to R_i.
    c. We rank $\{R_1, \ldots, R_n\}$ by their similarity with D, i.e. by $sim(D, R_i)$ and return that ranking to the user.

Note that in addition to returning the ranking, each similarity score can be *explained* by providing the (top-scoring) abstract paths that appeared in the sum in step 2b.

In the following two sections, we will give more details on each of the steps 1, 2a and 2b of this general algorithm.

E. Step 1: identify "good" abstract paths

Our first goal is to identify those abstract paths that are found often between dossiers and resources that form positive examples, but rarely between negative ones. We can interpret these abstract paths as "association rules" that predict the association between a dossier and a resource.

Similar to approaches in association rule mining, we therefore score an abstract path p by its *lift* (cf. e.g. [27], chapter 6), which is defined as the probability of p being observed within the set of all paths that connect positive examples, divided by the overall probability of observing p:

$$Lift(p) = \frac{P(p|positive)}{P(p)}$$

These probabilities are estimated by computing the relative frequencies of these events in the training data.

In our example from **Figure 1**, starting from the dossier "Feb 13", there are overall 5 paths that end in a resource: the direct assignment of report1, the two paths that lead via John Doe directly to the report1 and report2, and the two paths that also lead via John Doe, but first making the detour via the "Sales" topic and the "sales rep." role.

Now, let us consider the abstract path $p =$ (*dossier-has-creator, resource-has-author*). Since there are 5 paths overall and 2 of these paths correspond to *p*, the overall probability of observing this path is $P(p) = \frac{2}{5} = 0.4$. On the other hand, only "Feb 13" and "report1" are directly connected. There are 3 paths (including the direct assignment) that connect these two nodes. Of these, only one corresponds to *p*, hence the probability of this path being observed to connect a positive example is $P(p|positive) = \frac{1}{3} = 0.33$. Thus, we arrive at $Lift(p) = \frac{0.33}{0.4} = 0.83$. Any lift below 1 means that it is less probable to observe this path with positive examples than to observe it in general – we would hence discard p.

In practice, abstract paths are found by breadth-first-search: starting from each dossier D in a resource graph, we follow the edges, searching all paths that end in a resource. Each time we encounter a resource R, we record which abstract path corresponds to the (concrete) path that led us from D to R and increment the count(s) for that abstract path's probabilities.

In general, enumerating all paths between any two pairs of nodes in a graph, is an NP-hard problem and hence requires exponential time. Therefore, it is important to limit the depth of the search. It can also be useful to constrain the search by specifying that certain edge types should not appear more than once within any abstract path – something which can often be done without losing any relevant abstract paths.

F. Step 2a: identify candidate resources

After the identification of all significant paths, with a lift higher than 1, between the dossiers and the documents we can apply them to the other dossiers.
As depicted in **Figure 1**, there is a path *resource-has-author* between John Doe and the "report 1", which itself belongs to the dossier "Feb 13". Now let's assume that after all the calculations in Step 1, the lift of the *resource-has-author* path is significant. We can now infer new knowledge on **Figure 1** by applying the same path to the "report 2". Because of the

shared author John Doe, we can infer, that the "report 2" is to some extent related to the dossier "Feb 13". Therefore we instantiate a new relation between the dossier "Feb 13" and "report 2" called *is-related-to*.
This results in a new representation of dossiers and their relations to resources.

Our implemented prototype is developed in JAVA. Its backbone is the ArchiMEO enterprise ontology, which is capable to store the data and abstract paths. For each abstract path, we generate a rule, where the left side of the rule is an abstract path and the right side tells the reasoner to create a new edge that (tentatively) connects the dossier D to the candidate resource R, whenever there is a path between D and R that corresponds to the left side of the rule. The population of the ontology happens through the JENA API (http://jena.apache.org/) in a separate population module. This comprises an automated approach to walk through every item in the SEEK!SDM system and creating its instances in the ontology. Rules are manually defined and managed within an Ontology management tool called TopBraid (http://www.topquadrant.com). They are expressed as SPIN Rules and triggered in the prototype through the SPIN APIs (http://topbraid.org/spin/api/ and http://topbraid.org/spin/api/1.2.0/). When setting off the inferencing, the reasoner will insert new links that preliminarily connect dossiers to all candidate resources that can be reached via any of the abstract paths defined before. As explained in the next chapter, the similarity of these resources to the dossier can be ranked, by summing up the lifts of the corresponding abstract paths.

G. Step 2b: score resources by similarity

While we identify the candidate resources $\{R_1, \ldots, R_n\}$, we need to record which paths have led from D to each R_i because the scores of the corresponding abstract paths will help us to score the candidates.

If we ignore the direct (manual assignment) between dossier "Feb13" and "report1" in the example depicted in **Figure 1**, and start at dossier "Feb 13" there are two acyclic paths that lead to the resource "report 1".

- The first (via John Doe) corresponds to the abstract path $p_1 = ($*dossier-has-creator, resource-has-author*$)$.
- The second path leads via the topic "Sales", the role "sales rep" and "John Doe" and corresponds to the abstract path $p_2 = ($*dossier-has-topic*, *role-has-topic*, *login-has-role*, *resource-has-author*$)$

We now define the similarity between the dossier D and a candidate resource R_i as the sum of scores of all abstract paths that correspond to those paths that lead from D to R_i (note: if there is more than one path between D and R_i that corresponds to the same abstract path, then the score of that abstract path will be added twice):

$$sim(D, R_i) = \sum_{p' \cong p \in P(D, R_i)} score(p')$$

where $P(D, R_i)$ is the set of all paths between D and R_i.

In our example, the similarity between dossier "Feb 13" and "report 1" will be computed as $score(p_1) + score(p_2)$ and p_1 and p_2 can help to explain why "report 1" was retrieved.

VI. Evaluation Results and Discussion

A. Experimental setup

We applied our approach to an instance of the SEEK!SDM system[2]. Table 1 summarises the data set and its characteristics in terms of number of nodes of different types, whereas Table 2 shows the number of edges of the various types.

Node type	**Count**
Resources	1474
Dossier	208
Tag	262
Topic	63
User	21
Role	3678

Table 1. Number of nodes of different type in the studied data set

As can be seen, there is a large number of roles compared to users (logins) in the system. This is because the system automatically creates 'empty roles' for each resource, regardless whether the role is assigned to a user at the point of its creation or not. The roles are available to be assigned to users and user groups at any time later in the resource's life time.

Edge type	**Count**
Dossier-has-resource	1258
Dossier-has-topic	208
Tag assignments (Resource-has-tag, Dossier-has-tag)	6127
Resource-has-creator	1474
Dossier-has-creator	208
Login-has-role	247
Role-has-topic	63

Table 2. Number of edges of different type in the studied data set

Further, we see that each dossier belongs to exactly one topic in the topic tree. Similarly, each dossier and each resource has exactly one creator.

We configured out algorithm introduced above to discover all paths of length ≤ 4 leading from a dossier to a resource when executed but to ignore any paths that had more than one consecutive edges of type "*resource-has-creator*". This is for the following reason: if a path has for example two consecutive edges of type "*resource-has-creator*" this means that one resource is related to another resource via a person who has created both resources (consider the shortest path from "report1" to "report2" as shown in **Figure 1**). Hence, not restricting the paths would very quickly lead to the inclusion

[2] see http://www.sdm4you.ch/ for a demo version

of all resources since a few power users created the majority of resources.

B. Results

With these restrictions, our algorithm discovered roughly 4 million paths connecting dossiers to resources. About 88,000 of the paths represented a positive example (i.e. the paths connected a dossier to a resource that was manually assigned to that dossier).

Table 3 provides the abstract paths that were observed at least once with paths connecting positive examples and have a lift greater than 1, sorted by their lift (last column). The table also shows the support count of each abstract path, i.e. how often a corresponding path was observed in the data.

The first path discovered is trivial. The interesting paths discovered are paths nr. 1.2 and 1.3 – with a lift value significantly greater than 1. Starting from a dossier D, path 1.2 will recommend resources that share a tag with another resource that already belongs to D. Path 1.3 will additionally recommend resources that share a tag directly with D.

Nr	Abstract path	Support count	Lift
1.1	dossier-has-resource	1258	45.35
1.2	dossier-has-resource -> resource-has-tag -> resource-has-tag	196569	11.99
1.3	dossier-has-tag -> resource-has-tag	21406	6.45
1.4	dossier-has-tag -> dossier-has-tag -> dossier-has-tag -> resource-has-tag	28950	1.14
1.5	dossier-has-tag -> resource-has-tag -> resource-has-tag -> resource-has-tag	233910	1.14
1.6	dossier-has-creator -> resource-has-creator	38168	1.03

Table 3. Abstract paths discovered (with tags)

We also experimented with excluding tagging relations, i.e. edges of type "*dossier-has-tag*" and "*resource-has-tag*". This graph is much smaller, such that there are only roughly 2.5 million paths connecting dossiers and resources, 10,000 of them connecting positive examples. Nevertheless, this experiment revealed some additional interesting paths – all of these paths with lift greater than 1 are displayed in

Table 4.

For instance, path 2.2 will recommend resources that were created by the same person who created the input dossier D. Path 2.3 recommends resources that were created by a person who also created a dossier with the same topic as D. Finally, path 2.4 recommends resources that have been created by a person whose role is associated with the same topic as D. Since in the SEEK!SDM system access rights are tied to roles and are usually granted for areas of the topic tree (e.g. a sales representative has access to the topic "Sales and Marketing" in the tree in **Figure 1**, with all its sub-trees), this abstract path makes intuitive sense: resources created by a person with a role of a sales representative are likely to be somewhat similar to all dossiers in the "Sales and Marketing" area of the topic tree.

Nr	Abstract path	Support count	Lift
2.1	dossier-has-resource	1258	251.63
2.2	dossier-has-creator -> resource-has-creator	38168	5.72
2.3	dossier-has-topic -> dossier-has-topic -> dossier-has-creator -> resource-has-creator	459192	2.70
2.4	dossier-has-topic -> role-has-topic -> login-has-role -> resource-has-creator	23933	1.31

Table 4. Abstract paths discovered (without tags)

C. User Interface

Since a business user will not be interested in the technical background of a recommendation, recommendations for resources are explained in a user friendly way. Therefore the abstract paths on which the recommendations are based are translated into natural language. Table 5 below shows the user representation of the most interesting paths discovered.

Abstract path	Natural language description
dossier-has-resource -> resource-has-tag -> resource-has-tag	"the recommended resource shares a tag with a resource of this dossier"
dossier-has-tag -> resource-has-tag	"the recommended resource shares a tag with this dossier"
dossier-has-creator -> resource-has-creator	"the recommended resource was created by the person who also created this dossier"
dossier-has-topic -> dossier-has-topic -> dossier-has-creator -> resource-has-creator	The recommended resource was created by a person who also created a dossier with the same topic as this one.

Table 5. User-friendly description of abstract paths

D. Discussion

As observed above, the discovered paths make intuitive sense – i.e. none of these is really surprising in the sense that a human might not have come up with it, too. This is a valuable insight because it confirms that the analysis produces reasonable and comprehensible results.

However, the actual value of the analysis – and where it goes beyond what a human could do – lies in the fact that we are now able to quantify how reliably an abstract path will identify similar resources: for instance, path 2.4 in

Table 4 was mentioned as a "good rule" for identifying related resources by an expert previous to the automatic analysis. The analysis reveals, however, that other paths are significantly more reliable.

All in all, this shows that our analysis is able find reasonable paths (i.e. symbolic "rules" that make intuitive sense to experts and users). Scoring paths with their lift additionally enables us to use them in *ranking* recommended resources, as would be the case with a sub-symbolic approach.

VII. Conclusion

In many enterprises the structuring of file repositories remains subject of individual users leading to the well-known problems of multiple filing, overlooking relevant information and incomprehensible folder structures. Although information management systems allow for user-friendly handling of electronic documents of all types, its handling mainly copies paper filing. Within the SEEK!sem project new ways of supporting information management by applying semantic technologies and machine learning are researched. In this paper we presented an approach to improve current information systems by automatically identifying and recommending related information resources: by applying a rule-based similarity approach, the system can determine the reason for the similarity of resources. The recommendations are based on rules, exploiting mostly context similarity of information resources. Since the context of information objects is described in an enterprise ontology, it can be processed automatically. Using the context as a similarity base allows us to include different types of information resources, especially types with only limited processing and interpretation possibilities like images.

It is an important feature of our approach that it is based on a graph with labeled nodes and edges, i.e. one where specific classes of objects are foreseen and relations between them have a well-defined meaning. We exploit this meaning for explaining similarities to business users, i.e. the context of information objects is described not only in a machine understandable but also in a cognitively adequate representation for humans.

With the implementation and preliminary evaluation of the SEEK!Sem recommender described in chapters 0 and VI, the feasibility of discovering useful rules for recommending related information has been demonstrated. This process eases the handling of resources in information systems, prevents the emergence of information silos and thus increases the search ability and benefit of the system as a whole.

Acknowledgment

The SEEK!Sem research project is funded by the Commission for Technology and Innovation (CTI). Project KTI Nr. 14604.1 PFES-ES.

References

[1] The Open Group, “ArchiMate 2.0 Specification.” The Open Group, 2012.

[2] A. Hevner and S. Chatterjee, “Design Research in Information System - Theory and Practice,” in *Integrated Series in Information Systems*, S. Voß and R. Sharda, Eds. 2010.

[3] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: item-to-item collaborative filtering,” *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, Jan. 2003.

[4] Q. Lu and J. G. Conrad, “Bringing Order to Legal Documents - An Issue-based Recommendation System Via Cluster Association,” in *KEOD*, 2012, pp. 76–88.

[5] K. Al-Kofahi, P. Jackson, M. Dahn, C. Elberti, W. Keenan, and J. Duprey, “A Document Recommendation System Blending Retrieval and Categorization Technologies,” in *AAAI Workshop*, 2007, pp. 9–18.

[6] G. Salton, A. Wong, and C. S. Yang, “A vector space model for automatic indexing,” *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[7] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, “Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 355–369, 2007.

[8] E. Armengol and E. Plaza, “Using symbolic descriptions to explain similarity on CBR,” in *Proceedings of the 2005 conference on Artificial Intelligence Research and Development*, 2005, pp. 239–246.

[9] M. S. Michèle Sebag, “A Rule-Based Similarity Measure,” in *Lecture Notes in Computer Science*, Springer, 1994, pp. 119–131.

[10] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, “Large Scale Online Learning of Image Similarity Through Ranking,” *The Journal of Machine Learning Research*, vol. 11, pp. 1109–1135, Mar. 2010.

[11] E. Minkov and W. W. Cohen, “Learning to rank typed graph walks,” in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis - WebKDD/SNA-KDD ’07*, 2007, pp. 1–8.

[12] E. Minkov and W. W. Cohen, “Learning graph walk based similarity measures for parsed text,” in *EMNLP ’08 Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 907–916.

[13] E. Minkov and W. W. Cohen, “Graph based similarity measures for synonym extraction from parsed text,” in *Workshop Proceedings of TextGraphs-7 on Graph-based Methods for Natural Language Processing*, 2012, pp. 20–24.

[14] D. Fensel, S. Decker, M. Erdmann, and R. Studer, “Ontobroker: The Very High Idea,” in *Proceedings of the 11. International Flairs Conference (FLAIRS-98)*, 1998, no. May.

[15] H. Stuckenschmidt and F. Van Harmelen, “Ontology-Based Metadata Generation from Semi-Structured Information,” in *Proceedings of the 1st international conference on Knowledge capture*, 2001, pp. 163–170.

[16] V. Uren, P. Cimiano, S. Handschuh, M. Vargas-vera, E. Motta, and F. Ciravegna, “Semantic annotation for knowledge management : Requirements and a survey of the state of the art,” *World Wide Web Internet And Web Information Systems*, vol. 4, pp. 14–28, 2006.

[17] B. Thönssen, “Formalizing low - level governance instruments for a more holistic approach to automatic metadata generation,” in *Proceedings of the 5th International Conference on Methodologies, Technologies and Tools enabling e-Government*, 2011, pp. 1–12.

[18] B. Thönssen, “An Enterprise Ontology Building the Bases for Automatic Metadata Generation,” in *Proceedings of the 4th International Conference on Metadata and Semantics, MTSR1200*, 2010, pp. 195–210.

[19] A. Martin, “Linked Enterprise Models and Objects providing Context and Content for creating Metadata.” 2011.

[20] B. Thönssen, “Automatic, Format-independent Generation of Metadata for Documents Based on Semantically Enriched Context Information,” University of Camerino, 2013.

[21] K. Hinkelmann, E. Merelli, and B. Thönssen, “The Role of Content and Context in Enterprise Repositories,” in *Proceedings of the 2nd International Workshop on Advanced Enterprise Architecture and Repositories - AER 2010*, 2010.

[22] M. S. Fox, M. Barbuceanu, M. Grüninger, and J. Lin, “An Organization Ontology for Enterprise Modelling,” *Simulating Organizations: Computational Models of Institutions and Groups*, no. AAAI/MIT Press, pp. 131–152, 1996.

[23] M. Uschold, M. King, S. Moralee, and Y. Zorgios, “The Enterprise Ontology,” *The Knowledge Engineering Review*, vol. 13, no. Special Issue on Putting Ontologies to Use. AIAI, The University of Edinburgh, 1997.

[24] M. Leppänen, “A Context-Based Enterprise Ontology,” in *Proceedings of the EDOC International Workshop on Vocabularies, Ontologies and Rules for the Enterprise (VORTE’05)*, 2005, no. Lecture Notes in Computer Science, pp. 17–24.

[25] P. Bertolazzi, C. Krusich, M. Missikoff, and V. Manzoni, “An Approach to the Definition of a Core Enterprise Ontology : CEO,” in *International Workshop on Open Enterprise Solutions: Systems, Experiences, and Organizations - OES-SEO 2001*, 2001, pp. 104–115.

[26] G. L. Geerts and W. E. McCarthy, “The Ontological Foundation of REA Enterprise Information Systems,” 2000.

[27] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison Wesley, 2005.