Using Domain Ontologies for Finding Experts in Corporate Wikis

Ralph Schäfermeier
Freie Universität Berlin
Department of Mathematics and Computer Science
Corporate Semantic Web Group
Königin-Luise-Str. 24-26
14195 Berlin, Germany
schaef@mi.fu-berlin.de

Adrian Paschke
Freie Universität Berlin
Department of Mathematics and Computer Science
Corporate Semantic Web Group
Königin-Luise-Str. 24-26
14195 Berlin, Germany
paschke@mi.fu-berlin.de

ABSTRACT

Finding experts is a relevant problem in large, distributed organizations, and automated solutions are needed. In this paper, we propose an approach for finding experts among Wiki authors, since Wikis have emerged as important collaboration and knowledge management tool in enterprizes.

By analyzing revision histories and by semantically mapping Wiki contributions to concepts defined in corporate domain ontologies we identify experts. We apply semantic similarity metrics in order to detect references to ontology topics not explicitly mentioned in the text. Furthermore, we use information from the revision history in order to assess the level of expertise and examine the collaborative peer-reviewing processes happening in Wiki systems in order to calculate a reputation score for each author, based on the author's contribution lifetime.

We evaluated our approach on the Eclipse project Wiki and conducted a survey with Eclipse project members to assess the quality of our expert finding approach. The results show that the approach yields accurate expertise information.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models, search process*

General Terms

Algorithms, Experimentation, Measurement

Keywords

Enterprise Search, Expert Finding, Expertise, Wiki, Reputation, Semantic Web, Ontology, Information Retrieval

1. INTRODUCTION

Wikis have emerged as important collaboration and knowledge management tools in large, locally distributed communities and corporate environments. However, not all knowledge can be documented. Especially tacit knowledge which enables individuals to solve complex problems, and which is the result of personal experience and training, is hard to make explicit. Moreover, a number of studies conducted among employees in medium-sized and large companies suggest that people often prefer talking to an expert rather than referring to a document when they need help in coping with a task (cf. [19] and [17]).

Besides the above mentioned reasons, social aspects play an important role. A person can, for instance, act as an intermediary to other important persons by directly introducing the two parties to each other or by revealing information about the other person that helps lowering the barrier to initiate communication [17].

In general, a person with expertise can deliver more practical knowledge than a document can do because persons with expertise can apply their general experience to broader classes of problems, while documents tend to remain focused on a rather tight problem context or completely lack such context [34].

Most importantly, a person with expertise can adjust her or his vocabulary to that of the inquirer who may only have basic knowledge (or even none at all) about the problem domain and the appropriate terminology, making person-to-person knowledge transfer more efficient than document-to-person knowledge transfer.

The paper contributes with an approach for identifying experts among Wiki authors using formalized domain knowledge in the form of a light-weight domain ontology and develops a prototypical expert finder system. The main contribution consists in the fact that the presented approach is fully unsupervised and does not depend on the initial presence of external expertise information. Furthermore, by using domain ontologies and considering the semantic distance between domain concepts, we are able to expand user queries so as to to include experts in similar or related topics in the search result.

The approach has been tested and evaluated using the project Wiki of the Eclipse Foundation[1] and the Software Engineering Ontology (SEOntology) developed at the Digital Ecosystems and Business Intelligence Institute (DEBII), Curtin University of Technology, Perth, Australia [31].

In the following sections, we outline two use cases for our system. In section 3, we give an overview of existing approaches to the problem of expert finding. In section 4, we present our expert model based on Wiki contributions. In section 5, we present the re-

I-SEMANTICS 2011, 7th Int. Conf. on Semantic Systems, Sept. 7-9, 2011, Graz, Austria

[1] `http://wiki.eclipse.org/Main_Page`

sults of the evaluation, followed by a discussion about the validity of our results. Section 6 concludes with an outlook on future work.

2. USE CASES

Use cases for expert finding range from need for ad hoc support where experts help solving specific problems to recruitment tasks where experts are sought for long-term cooperation. In this section, we describe a concrete use case for each of the two problem classes.

Finding experts for a project: Finding the right people to work in a project is a problem especially for large, locally distributed organizations or projects where project heads lose track of their employees' skills. By choosing the wrong team members, organizations fail to take full advantage of their human capital. Employees reveal information about their expertise by the artifacts they produce during their work. In domains with knowledge intensive work, these artifacts consist of documentation in the most part. Automatic expert finding can leverage these artifacts in order to identify optimal candidates whose expertise suits the requirements of a project.

Issue tracking: Issue tracking systems automate the process of handling support requests by customers. Customers can use the system to report problems with a service or product. If the issue is known, it can be resolved by a first level support co-worker. Otherwise, the issue is escalated to higher support levels and will be worked on by experts. After resolving the issue, the expert adds documentation about the issue and the necessary steps to resolve it to a knowledge base, by which the previously new issue becomes a known issue, enabling first level support to use this information in the future. In a number of systems, the bulk of the work done by first level support has been shifted to the customers, leaving it to them to find a solution to a problem in the knowledge base or to classify the problem and file a support ticket. Since customers are not experts, it is possible that they classify their issue using the wrong category. In this case, the incident is assigned to the wrong expert who has to reclassify the issue and forward the ticket, placing additional work load on the expert.

Automated expert selection can be applied in order to take the burden of problem classification from the customers and the experts, resulting in less response time. The knowledge base can be used for extracting expertise evidence for known topics. With the aid of domain vocabulary and information about topic similarity, expertise information can be derived for topics which are not explicitly included in the knowledge base.

3. RELATED WORK

In the following section, we give a brief overview of existing approaches tackling the problem of automated expert finding. These can be divided into five groups:

1. manual approaches where the experts themselves provide information about their own skills and expertise
2. automatic and semi-automatic approaches that use information-mining techniques on document corpora
3. automatic and semi-automatic approaches based on the analysis of communication patterns and social networks
4. automatic approaches based on the analysis of revision histories
5. unified document and people search

Many works in this area are based on a combination of the above-mentioned approaches.

3.1 Manual Approaches

Hand-crafted expert databases are used as in-house solutions in companies or as publicly accessible services in the World Wide Web. The latter are predominantly employed in an academic or corporate context or act as a bridge between the two. Examples for such services are *ProfNet*[2], a service that acts as agent between journalists and experts from the academic domain [9], *COS Expertise*[3], or the *Virginia Tech Expert Database (VTED)*[4].

While easy to implement, the manual maintenance of expertise information imposes several disadvantages. The most obvious is the additional work that it puts on the experts. As the willingness of experts to provide information varies, so do the resulting expertise data.

A second problem is to make self-assessed expertise comparable. In team building or support ticket routing scenarios, it might be essential to pick the best expert and not just any. In order to compare her own expertise with that of someone else on a reliable scale, an expert would have to know about everyone else's expertise in the system.

A third problem is that self-assessed expertise tends to be biased, either consciously or unconsciously. The motivation for skewing expertise information about oneself is twofold. On the one hand, as observed by Fazel-Zarandi and Yu [18], people tend to exaggerate when providing expertise information about themselves with the aim of gaining higher reputation and possibly a promotion or other advantages. On the other hand, Desouza observed in a study conducted among 50 software developers that people may also be inclined to understate their expertise, either to simply avoid additional workload or to avoid being tied to a certain role. One respondent brings it to the point: "...soon I will be dubbed the 'Unix Guru' and that is all I will end up being in charge of..." [15].

3.2 Mining Expertise Information from Text Corpora

Approaches for finding experts by mining expertise information from text are based on the principle of interpreting co-occurrences of terms relevant to the problem domain and names of persons by the means of probabilistic models. If a person is identified frequently in the context of terms describing a domain, then this person is considered as an expert in that domain. An example of such an approach was described by Balog [5].

A disadvantage of this approach is that it does not identify experts based on artifacts produced by them but rather by artifacts about them and produced by others. While these approaches yield reliable results, they only capture expertise information about people who have already gained a sufficiently high degree of reputation, such that other people write about or cite them.

An approach proposed by Jung et al. [21] uses data from the Open Archives initiative [5] in order to find experts based on co-authorship in academic publications. They use linguistic analysis and keyword matching for identifying topics in publications and associate them to authors accordingly. Open Access data is also used to associate topics with institutions, thus allowing an expert seeker to identify research institutions with expertise in a particular topic, rather than a single person. Unlike our approach, this approach is focused on the academic field.

[2]`http://www.profnet.com`
[3]`http://expertise.cos.com/`
[4]`http://www.research.vt.edu/vted/`
[5]`http://www.openarchives.org/`

3.3 Analysis of Communication Patterns

Expert finding by analyzing communication patterns is based on the assumption that experts communicate more than average with other experts in the same field. The first approach in this domain was proposed by Schwartz et al. who analyzed email communication between 15 different sites [29]. A similar approach was proposed by Campbell et al. [10] who added cluster analysis in order to capture not only who is related to who but also on which topics people communicate.

In the initialization phase, algorithms that follow this approach must be seeded with an initial set of known experts. Another problem with this type of approach are privacy concerns. While published documents concentrate on a particular topic and are by nature intended to be accessible by a certain public, email communication is generally presumed to be private.

3.4 Finding Experts in Wikis or Other Social Networks

Demartini proposes an algorithm similar to the one presented in this paper for finding experts in Wikipedia [14]. In contrast to our work, Demartini's algorithm does not assess the community's trust in a potential expert's content by considering its longevity, instead, it uses the number of links pointing to a page covering a particular topic. The problem with this approach, as Demartini pointed out, is that link count can be interpreted as a measure for topic popularity rather than for author reputation. An evaluation of the approach is not available.

Yang et al. also exploit the link structure in Wikipedia articles for the expert finding task [33]. Unlike our approach, Yang et al. only use Wikipedia in order to establish a taxonomy of possible topics, while the experts themselves are identified using publication data. Accordingly, the application scenario envisioned in this work is finding experts for reviewing scientific papers.

Stankovic et al. propose an approach exploiting user traces in the Linked Open Data cloud for identifying experts [30]. Their approach relies on expertise hypotheses expressed in RDF, each hypothesis fitting a particular kind of user activity traceable in the LOD. The authors state that their approach is sensitive to LOD data quality, and intensive manual data cleaning had to be performed in order to achieve reasonable results.

3.5 Analysis of Revision Histories

A number of works exist that use information from revision histories of source code management systems for finding experts among software developers.

Mockus et al. propose a system called *Expertise Browser* [25] which enables developers to identify the person who is responsible for particular software modules while browsing through the source code repository. A similar approach was taken by Alonso et al. who proposed a system that identifies relevant key words in bug requests and maps them to developers potentially able to fix the bug [3]. They use domain vocabulary which consists of the software module names. Another work in this field is the approach proposed by Matter et al. which uses lexical matching of terms in bug requests to terms used in source code documentation or class, method, and variable names [24].

Other works examine the revision history of document management systems and derive expertise information based on user activity.

Nasirifard and Peristeras presented an approach for identifying experts among users of document centric collaborative platforms [26]. They implemented a system that captures user activity in a BSCW using the revision history and assigns expertise to people who work on particular documents. Unlike our approach, the proposed system only considers activities on the document level. It does not analyze the internal structure of a document nor does it recognize single contributions within a document.

3.6 Unified Document and People Search

A number of hybrid approaches have been proposed that retrieve documents as well as related persons with potential expertise in the topic covered by the retrieved documents and present both side-by-side in a mashup.

Amitay et al. make use of social network information available from Web 2.0 applications, such as Blog comments, user discussions, and tag systems [4]. They apply a standard vector space model for document retrieval and then establish associations between the retrieved documents and persons who have contributed to them and similar documents.

4. EXPERT MODEL

The expert model developed in this work is grounded in the cognitive model of experts described by Bransford [8]. According to Bransford, experts store knowledge in a way that enables them to recognize basic structures and relevant dimensions of a problem, enabling them to find a more efficient approach to solve the problem than laymen. Likewise, experts are able to identify and name abstract concepts from the problem domain using domain specific terminology. The underlying assumption in this work is that Wiki users who are experts in a specific topic write about this topic, either by using topic specific vocabulary, or by contributing content to topic specific Wiki sections denominated using corresponding vocabulary. In our model, we utilize the employment of domain specific vocabulary as expertise indicators.

Furthermore, as Chi pointed out [12], experts in a certain topic generally tend to possess expertise in similar or related topics as well. For example, it can be assumed that a person with expertise in UML class modeling has at least some expertise in use case modeling, too. On the other hand, one cannot assume that each person publishes content about every topic in which he has expertise, and this information can easily be missed. Therefore, our model incorporates ontological knowledge about semantic relations between concepts of the problem domain, in order to capture expertise evidence about topics which are not explicitly referred to in a Wiki author's contributions.

This may also be beneficial from the inquirer's point of view. As Kuhlthau points out, finding the appropriate search terms for a query poses a problem for information seekers interacting with an information retrieval system [22]. Taking into account similar concepts may increase the recall level even for imprecise queries.

In the following sections, we describe how we calculate a topical expertise score for Wiki authors based on their Wiki contributions. We then depict how we enhance our expert model with the aid of semantic relations and ontology based similarity metrics in order to capture implicit expertise evidence for topics not mentioned in the Wiki content. Finally, we show how we improve the expertise score by weighting it with an author reputation score.

4.1 Expertise Score Based on Wiki Contributions

An expertise score is a figure revealing how much expertise an individual has in a given topic. Thus, expertise e is a function $a \times t \mapsto e$ for a given author a and a given topic t. Since expertise is hard to quantify, and a sheer number does not reveal any feasible information, we are rather interested in comparing expertise scores of different individuals with each other and ranking them accordingly.

Thus, the only requirement for an expertise score is that it must be higher for individuals with more expertise and lower for individuals with less expertise in the same topic.

We assume that an individual who contributes content relevant to a specific topic to the Wiki has expertise in this topic.

In contrast to traditional documents, Wiki pages are not authored by a single person. Instead, many different authors contribute to a page by adding, deleting, or moving content. Each time an author edits a page, a new revision of this page is created. Wiki systems maintain a revision history with detailed information about what has changed, the originator of the changes, and the order in which the changes were made. Our approach uses this information for assigning authorship to Wiki content on a very detailed level.

Every wiki page has a title that describes the topic the page is about. Additionally, each page can be broken down into subsections of different levels, witch each subsection having a title as well. It can be assumed that each subsection deals with a specialized subtopic of the page's topic, and the level of specialization is related to the section level. Assuming section levels start with one, we can, for the sake of simplicity, consider pages as super sections with level zero.

Based on these observations, we calculate a simple contribution based expertise score as follows:

$$expertise_{simple}(a,t) := \sum_{s \in S_{a,t}} weight_s(level(s)) + \sum_{w \in W_{a,t}} weight_w(w) \tag{1}$$

where by $S_{a,t}$ we denominate the set of sections that cover the topic t and under which author a has contributed content.

$weight_s$ is a weighting factor depending on the section level. We used a simple milestone metric in order to express the relevance of a section according to its level. The underlying assumption is that a section with a higher level is about a more general topic, and contributions to a highly specialized topic should be reflected with a higher weighting in the expertise score.

Accordingly, by $W_{a,t}$ we denominate all occurrences of terms contributed by author a that can be mapped to topic t, weighted by a relevance function $weight_w$. We applied the tf-idf weight in order to capture term relevance w. r. t. the document.

The mapping of terms to topics in the problem domain is accomplished by lexical mapping of term occurrences either in the text or in page or section titles to the lexical description of concepts in the ontology by `rdfs:label`[6] annotations. We apply part-of-speech tagging and word stemming and only consider the stems of nouns and verbs.

The advantage of this mapping is that only terms relevant to the problem domain are taken into account, reducing the amount of features to be considered when calculating the expertise score.

4.2 Using Ontological Relations for Identifying Implicit Expertise Evidence

As stated in the introduction to this section, our expert model captures information about potential expertise in topics not explicitly covered in Wiki contributions by the means of semantic relations expressed in the ontology.

While in the first step, we used lexical term-concept matching in order to purge irrelevant terms, we now use concept relations in order to detect similar topics and add them to the set of relevant features. We utilize the class hierarchy established by the `owl:subClassOf`[7] OWL property and other selectable subtypes of `owl:ObjectProperties`[8] in order to capture concept relatedness.

[6] `http://www.w3.org/TR/rdf-schema/#ch_label`

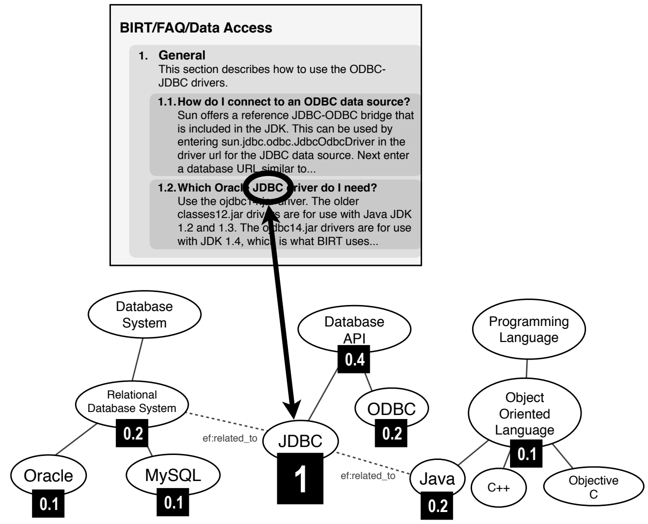

Figure 1: Weighting of detected and related features using ontological similarity measures

Based on these relations, we calculate a relevance score using ontology based similarity measures until a defined threshold is reached (see figure 1). All concepts with a similarity value higher than this threshold are considered similar and added to the feature set, weighted by its similarity value.

Several graph- and entropy-based concept similarity measures have been proposed in the literature, such as a simple measure based on path lengths by Leacock and Chodorow [23], a slightly more complex graph-based measure by Wu and Palmer [32] which also takes into the account concept depth and the distance to the least common subsumer, and entropy based measures such as that proposed by Resnik [28] which defines the similarity of two concepts as the entropy of the least common subsumer, or the one proposed by Jiang and Conrath [20] which also takes into account the specificity of each of the two concepts and thereby their difference.

In our prototype, we implemented the above-mentioned four similarity measures. It turned out that the impact of the similarity measure was insignificant wrt. to the outcome. Therefore, we used the rather simple graph-based measure by Wu and Palmer because of its low computational complexity.

This yields a consolidated expertise score which is calculated as follows:

$$expertise(a,t) := \sum_{\substack{t_{sim} \in T, \\ sim(t,t_{sim}) \geq sim_{min}}} expertise_{simple}(a,t_{sim}) \cdot sim(t,t_{sim}). \tag{2}$$

Even if topic t is never referenced by author a, but neighbouring topics t_{sim} with a similarity to t greater than the threshold sim_{min}, then t benefits from a's expertise in each topic t_{sim}. The more similar t and t_{sim} are, the higher the benefit for t.

4.3 Considering Author Reputation

According to Ehrlich, reputation is another important factor when

[7] `http://www.w3.org/TR/owl-ref/#subClassOf-def`
[8] `http://www.w3.org/TR/owl-ref/#ObjectProperty-def`

it comes to judging experts [16]. In this work, the reputation of an author with respect to a certain expertise topic is assessed using the quality control processes accomplished by peer Wiki users. When authors contribute content to the Wiki, their contributions become subject to a perpetual revision process by other Wiki authors. This process is reflected in the Wiki's revision history. The Wiki principle encourages authors to change or even delete passages they object to. On the other hand, if a reviser considers a contribution to be relevant and correct, he can decide to keep or restore it if it has been deleted before. The contributions that survive over time can be considered public consensus. The expertise model developed in this work takes this revision process into account by considering a person the more reputable the more revisions her contributions have survived, i. e. the more persons presumably have agreed with what the person has contributed.

Algorithms calculating a reputation score for Wiki authors have been proposed by Chatterjee et al. [11] as well as by Adler and de Alfaro [2]. Their approaches cannot, however, be directly applied to the problem of judging experts, because they calculate either trust scores for isolated text paragraphs or an overall reputation score for authors, regardless of the topics the authors have contributed to. For judging experts, a reputation score is needed that assesses the reputation of each author w. r. t. each topic she has contributed to.

Therefore, we define the reputation of an author a w. r. t. a topic t as a function of the average lifetime of all of a's contributions to t.

A simple but naïve approach would be to relate reputation proportionally to contribution lifetime. However, this approach does not take into account the fact that certain passages might get deleted after a comparably short amount of time because they have become obsolete and their informational value has decayed. In contrast, other passages might describe more universal facts that may never become obsolete. This would skew the reputation score in favor of those authors who contribute more of the latter kind of facts to the Wiki. To circumvent this problem, we use a trust function with a slope that is steep at the beginning and flattens with increasing contribution lifetime. This reflects the observation that contributions become "established" once they have survived a certain amount of revisions. Therefore, we conceive trust in a contribution as the probability with which it remains untouched.

As an analysis of the Eclipse Project Wiki revealed, the majority of contributions that had survived four or more revisions were not deleted later on and could thus be considered stable.

The trust function we use in this work is defined as follows and reflects this observation (see fig. 2):

Assuming a single contribution c has survived the number of Δr_c revisions before it was deleted or, if it has not been deleted, until the current point in time, then we define the trust coefficient for this contribution c as

$$trust(c) := 1 - k^{-\Delta r_c}. \tag{3}$$

The higher the constant k, the more revisions are necessary to increase the trust index, while the latter never exceeds 1.

A potential problem with trust as a function of revision count is that an author can fraudulently boost the trust score of his contributions by consecutively creating new page revisions. To avoid this, our prototype identifies consecutive revisions by the same author and treats them as a single revision.

Assuming $C_{a,t}$ is the set of all contributions of author a to topic t, we define the reputation of a w. r. t. t as the average of the trust indices for all $c \in C_{a,t}$:

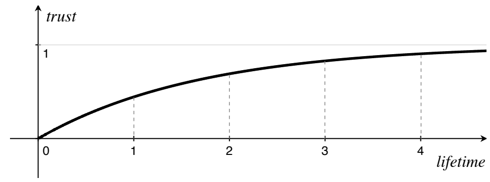

Figure 2: Growth of Trust in a User Contribution over Time

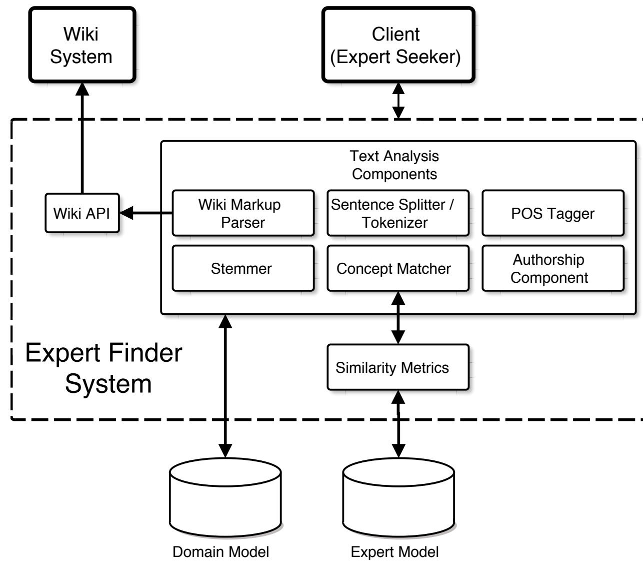

Figure 3: Architecture of the ExpertFinder System

$$reputation(a,t) := \frac{\sum_{c \in C_{a,t}} trust(c)}{|C_{a,t}|} \tag{4}$$

In order to incorporate author reputation into the expertise ranking, we weight the expertise score for an author a and a topic t by a's reputation score w. r. t. t:

$$expertise_{rep}(a,t) := expertise(a,t) \cdot reputation(a,t). \tag{5}$$

5. EVALUATION

For the evaluation, we implemented a prototypical system. The high-level architecture of the system is depicted in figure 3.

The Wiki API constitutes the interface to the Wiki system and provides operations for retrieving Wiki content and meta information about Wiki pages, revisions, and authors.

Once retrieved, the text of each revision is passed to the Wiki markup parser which identifies sections, links, and other structural markup like enumerations, tables, and info boxes. The remaining plain text undergoes sentence splitting, tokenization and part-of-speech tagging. Verbs and nouns are stemmed and passed to the concept matcher component which associates identified concepts from the problem domain to their occurrences in the text. The authorship component implements an advanced diff algorithm that attributes authorship to text passages with high accuracy and is resistant to text dislocation as well as vandalism.

5.1 Setup

We evaluated our approach on the Eclipse Project Wiki which is used by members of the Eclipse foundation mostly for project planning and documentation.

The wiki exists since 2005. For our evaluation, we analyzed Wiki contributions covering the period from 2005 to end of 2009. At the time of our analysis, the Wiki contained 10497 pages, edited by 3377 authors during 153882 revisions.

As a conceptual domain model we used the SEOntology [31] which models 364 concepts from the Software Engineering domain, based on the 2004 edition of the IEEE Software Engineering Body of Knowledge (SWEBOK)[9] (cf. [7]) and Ian Sommerville's Software Engineering textbook[10].

We prepared a survey and asked Wiki authors to assess their degree of expertise in 25 topics from the SEOntology on a scale reaching from *very high* to *(almost) none*. We also asked the participants to state whether they are a specialist in the particular topic or if the topic belongs to a rather general area where the participant has expertise in.

We selected the topics by information need as stated by Google Insights[11]. For each topic defined in the SEOntology, we queried Google Insights using as search terms the topic name and *"Software Engineering"* in order to disambiguate context. Then we selected the 25 topics with the highest ranking.

The Eclipse Project Wiki is based on the MediaWiki system[12] which provides no structured means for storing author contact information. Therefore, we had to collect contact information manually. We were able to gather contact information for the most active 200 authors who have together contributed 80 per cent of the entire Wiki content, measured in characters. 37 out of these 200 authors completed the survey, which corresponds to 1.1 per cent of all active authors.

In order to measure the effect of feature vector extension with similar topics and author reputation weighting, we prepared four different data sets with the following configurations:

(a) feature vector extension and author reputation weighting both disabled

(b) feature vector extension disabled and author reputation weighting enabled

(c) feature vector extension enabled and author reputation weighting disabled

(d) both features enabled

For calculating similarity of ontological concepts, we used the conceptual similarity measure proposed by Wu and Palmer [32], which is a path based metric but takes into account concept depth, yielding higher similarity values for proximate concepts with a high degree of specialization than for more general concepts. We found that the use of more complex (e.g. entropy-based) measures has no significant impact on the outcome (see section 4.2).

With each configuration, we ran a complete analysis and calculated the expertise score for each user and each topic.

[9] http://www.computer.org/portal/web/swebok/

[10] http://www.cs.st-andrews.ac.uk/~ifs/Books/SE7/index.html

[11] http://www.google.com/insights/search/

[12] http://www.mediawiki.org/

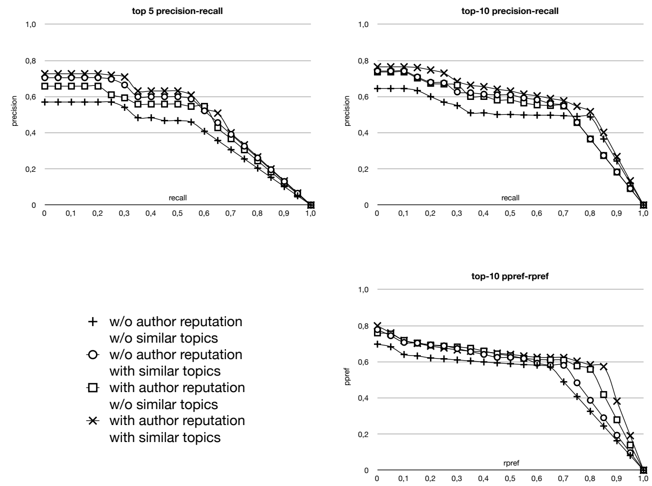

Figure 4: top-10 precision, top-5 precision, and top-10 preference precision

5.2 Evaluation Results

In order to evaluate the expertise data calculated by our system, we queried our system with each of the 25 topics and compared the result lists to those attained by the survey. Since the participants could only give answers in the range from zero to four, an unambiguous ranking is impossible. Calculating all possible rankings with ambiguous expertise scores can lead to a computing time of O(n!) in the worst case. Therefore, we picked 10000 random permutations between persons with equal expertise.

For each of the 25 topics and 10000 permutations, we calculated the precision and recall values and averaged them afterwards. Then, we calculated the top-5 and top-10 precision values for the ten standard recall levels.

Since the data attained by the survey are sparse, we additionally used the preference precision *(ppref)* measure which relies on the ranking order of retrieval results rather than on their absolute position in the result list and is therefore more robust against incomplete data [13].

Fig. 4 illustrates average top-5 precision, top-10 precision and top-10 preference precision.

The figures show that configuration a with both feature vector extension and author reputation weighting enabled leads to the best results. Particularly for high recall values, the precision value can be maintained at a relatively high level.

The fact that not only recall is positively affected but also precision may indicate that the approach of taking similar, not explicitly mentioned concepts into account balances the results in a positive manner. This supports the assumption that people who write about topics and thereby reveal their expertise also have expertise in similar topics.

Table 1 shows a comparison of our results to those described in section 3. Most of the evaluation results are available in the proceedings of the Enterprise Track at the Seventeenth Text REtrieval Conference (TREC 2008) [6].

5.3 Discussion

The results from our evaluation suggest that the algorithm yields accurate expertise data, especially with author reputation weighting and feature vector extension with the aid of ontological knowledge enabled.

Identifier	Approach	MAP	Corpus
Our approach	**automated**	**0.5160**	**EclipsePedia**
UvA08ESweb	automated	0.4490	TREC2008
ICTI3Sexp01	automated	0.4214	TREC2008
uogTrEXfeNPC	automated	0.4126	TREC2008
FDURoleRes	automated	0.4114	TREC2008
Yang et al.	automated	0.39	Wikipedia
THUPDDlchrS	automated	0.3846	TREC2008
WHU08NOPHR	automated	0.3826	TREC2008
utqurl	automated	0.3728	TREC2008
UCLex04	automated	0.3476	TREC2008
DERIrun3	automated	0.2619	TREC2008
LiaIcExp08	manual	0.2513	TREC2008
pristask204	manual	0.0977	TREC2008

Table 1: Comparison of our results to previous work by Mean Average Precision

However, there are factors putting into question the validity of our results.

One problem that can influence the outcome of our algorithm is participation inequality in Wiki systems and online communities in general. Looking at Wikipedia, Ortega et al. found that only a small percentage of users accounted for most of the contributions to the encyclopedia, while most of the users contributed little to nothing [27]. A similar participation pattern can be observed in the Eclipse project Wiki: While six per cent of the registered authors contributed 80 per cent of Wiki content (measured in character count), 94 per cent of the users account for the remaining 20 per cent of content.

The way our algorithm is crafted, it is prone to miss expertise evidence for authors with a low participation record. Because we recruited participants for our survey only from the group of highly active authors, our results may be biased to that effect. However, utilizing the preference precision measure which is more robust towards this effect yielded similar results.

The comparison with the results of other approaches is difficult, since it is based on different text corpora. The W3C corpus[13] used in the context of the TREC enterprise track has become a de facto standard for evaluating expert finder systems. However, since the W3C corpus consists of web documents and emails, it is not suitable for evaluating our approach which depends on wiki revision history data.

Another issue is that we based our evaluation on self-assessed expertise information. This approach is fault-prone as we discussed in section 3.

For more significant results, and in order to determine the effective usefulness of our approach, a user-based evaluation is necessary, assessing the ability of identified experts to respond to inquiries from a user's point of view.

It also remains to be investigated how the results would differ if our algorithm was applied to a closed enterprise Wiki. While the EclipsePedia has been selected for our evaluation because it is a professional project Wiki, social aspects may become more prevalent in closed enterprise Wikis where the mere presence of an expertise detection system could influence user behaviour significantly. E.g., an expert finding system can constitute an incentive for co-workers to contribute more knowledge to the wiki in order to get promoted or could scare off wiki authors who want to avoid additional workload.

[13] http://research.microsoft.com/en-us/um/people/nickcr/w3c-summary.html

6. CONCLUSION AND FUTURE WORK

In this paper we contributed with an approach for finding experts among Wiki authors by analyzing revision histories and using semantic background knowledge from domain ontologies. We implemented a prototypical semantic expert finding system and evaluated our approach on the Eclipse Project Wiki, using the SEONtology covering the domain of software engineering. Using domain vocabulary and concept relations defined in the ontology for detecting implicit references to concepts which are explicitly mentioned in the text, we were able to achieve a significant improvement of the retrieval results.

One of the shortcomings of our approach is that it is only suitable for enterprises or projects that use a Wiki for collaborative knowledge management. Since Wiki systems become more and more popular in enterprises, this issue will become less significant to a certain extent but will remain a limiting factor wrt. practical application.

Future work will focus on how the approach scales to Wikis with a broader topical focus and a user-based evaluation of the system's retrieval results.

In our first prototype, we use simple term lemma matching for identifying relevant topics in Wiki content. In order to apply our approach to a more general domain, proper entity recognition will probably play a more important role. We will have to implement a more robust entity recognition method, such as Open Calais[14], GATE[15], or Stanbol[16].

Furthermore, while our one-dimensional expert model is sufficient for identifying and comparing experts, there are cases where it would be desirable to leverage the different dimensions it is constructed from. We are therefore working on extending our model to a multidimensional expertise object model which allows to answer further questions about a person's expertise, like for example:

- How high is the person's reputation score w. r. t. to a specific topic?
- How high is the person's overall reputation score?
- How recent are the person's contributions to the topic in question?

We are planning to build our expertise model using OWL with the aim to make it interoperable with existing reputation models, thereby making our approach combinable with other approaches to expert finding or reputation assessment in general.

7. ACKNOWLEDGMENTS

This work has been partially supported by the "InnoProfile-Corporate Semantic Web" project funded by the German Federal Ministry of Education and Research (BMBF) and the BMBF Innovation Initiative for the New German Länder - Entrepreneurial Regions.

8. REFERENCES

[1] M. Ackerman, V. Pipek, and V. Wulf, editors. *Sharing expertise: Beyond knowledge management*. MIT press, 2003.

[2] B. Adler and L. de Alfaro. A content-driven reputation system for the Wikipedia. In *Proceedings of the 16th Intl. World Wide Web Conference (WWW 2007)*, 2008.

[14] http://www.opencalais.com/

[15] http://gate.ac.uk/

[16] http://incubator.apache.org/stanbol/

[3] O. Alonso, P. Devanbu, and M. Gertz. Expertise identification and visualization from CVS. In *MSR '08: Proceedings of the 2008 international working conference on Mining software repositories*, 2008.

[4] E. Amitay and S. Yogev. Finding people and documents, using web 2.0 data. In K. Balog, editor, *The SIGIR 2008 Workshop on Future Challenges in Expertise Retrieval (fCHER)*, volume 42, pages 46–52, 2008.

[5] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006.

[6] K. Balog, P. Thomas, N. Craswell, I. Soboroff, P. Bailey, and A. De Vries. Overview of the trec 2008 enterprise track. Technical report, National Institute of Standards and Technology, 2008.

[7] P. Bourque and R. Dupuis. Guide to the Software Engineering Body of Knowledge, 2004 Version. *IEEE Computer Society Professional Practices Committee*, 2005.

[8] J. Bransford. *How people learn: Brain, mind, experience, and school*, chapter 2, pages 31–50. National Academy Press, 2003.

[9] J. Brinkmann and D. Forbush. Introducing A Country To Internet-Enabled Expert Networks. *Communications of the ACM*, 46(11):127, 2003.

[10] C. Campbell, P. Maglio, A. Cozzi, and B. Dom. Expertise identification using email communications. In *Proceedings of the twelfth international conference on Information and knowledge management*, 2003.

[11] K. Chatterjee, L. Alfaro, and I. Pye. Robust content-driven reputation. *AISec '08: Proceedings of the 1st ACM Workshop on AISec*, 2008.

[12] M. Chi. *Two approaches to the study of experts' characteristics*, pages 21–30. Cambridge University Press, 2006.

[13] J. De Beer and M. Moens. Rpref: a generalization of Bpref towards graded relevance judgments. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, page 638. ACM, 2006.

[14] G. Demartini. Finding experts using wikipedia. In *Proceedings of the Workshop on Finding Experts on the Web with Semantics (FEWS2007) at ISWC/ASWC2007, Busan, South Korea*, 2007.

[15] K. Desouza. Barriers to effective use of knowledge management systems in software engineering. *Communications of the ACM*, 46(1):99–101, 2003.

[16] K. Ehrlich. *Locating expertise: Design issues for an expertise locator system*, pages 137–158. In Ackerman et al. [1], 2003.

[17] T. Erickson and W. Kellogg. *Knowledge communities: Online environments for supporting knowledge management and its social context*, pages 299–326. In Ackerman et al. [1], 2003.

[18] M. Fazel-Zarandi and E. Yu. Ontology-Based Expertise Finding. In *Proceedings of the 7th International Conference on Practical Aspects of Knowledge Management*, pages 232–243. Springer, 2008.

[19] M. Hertzum and A. Pejtersen. The information-seeking practices of engineers: searching for documents as well as for people. *Information Processing and Management*, 36(5):761–778, 2000.

[20] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics (ROCLING X)*, pages 19–33, Tapei, Taiwan, 1997.

[21] H. Jung, M. Lee, I. Kang, S. Lee, and W. Sung. Finding topic-centric identified experts based on full text analysis. In *2nd International ExpertFinder Workshop at ISWC*, pages 56–63, 2007.

[22] C. C. Kuhlthau. Inside the search process: Information seeking from the user's perspective. *Journal of the American Society for Information Science*, 42:361–371, 1991.

[23] C. Leacock and M. Chodorow. Combining local context and WordNet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

[24] D. Matter, A. Kuhn, and O. Nierstrasz. Assigning Bug Reports using a Vocabulary-Based Expertise Model of Developers. In *MSR '09: Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories*, pages 131 – 140, 2009.

[25] A. Mockus and J. Herbsleb. Expertise browser: a quantitative approach to identifying expertise. In *Proceedings of the 24th International Conference on Software Engineering*, pages 503–512. ACM New York, NY, USA, 2002.

[26] P. Nasirifard and V. Peristeras. Expertise extracting within online shared workspaces. In *Proceedings of the WebSci'09: Society On-Line*, 2009.

[27] F. Ortega, J. Gonzalez-Barahona, and G. Robles. On the inequality of contributions to Wikipedia. *Proc. HICSS 2008*, 2008.

[28] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[29] M. Schwartz and D. Wood. Discovering shared interests using graph analysis. *Communications of the ACM*, 36(8):78–89, 1993.

[30] M. Stankovic, J. Jovanovic, and P. Laublet. Linked data metrics for flexible expert search on the open web. In G. Antoniou, M. Grobelnik, E. Simperl, B. Parsia, D. Plexousakis, P. De Leenheer, and J. Pan, editors, *The Semantic Web: Research and Applications*, volume 6643 of *Lecture Notes in Computer Science*, pages 108–123. Springer Berlin / Heidelberg, 2011.

[31] P. Wongthongtham, E. Chang, and T. Dillon. Software Design Process Ontology Development. In *Proceedings of the 2nd IFIP WG 2.12 and WG 12.4 International Workshop on Web Semantics (SWWS) in conjunction with OTM 2006*, pages 1806–1813, 2006.

[32] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics Morristown, NJ, USA, 1994.

[33] K. Yang, C. Chen, H. Lee, and J. Ho. Efs: Expert finding system based on wikipedia link pattern analysis. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 631–635. IEEE, 2008.

[34] D. Yimam-Seid and A. Kobsa. *Expert-finding systems for organizations: Problem and domain analysis and the DEMOIR approach*, pages 327–358. In Ackerman et al. [1], 2003.