From Community towards Enterprise – a taxonomy-based search for experts

Gerald Eichler*, Andreas Lommatzsch†, Thomas Strecker†,
Danuta Ploch†, Conny Strecker†, Robert Wetzker†

*Innovation Development
Deutsche Telekom AG, Laboratories
Deutsche-Telekom-Allee 7
D-64295 Darmstadt
gerald.eichler@telekom.de

†DAI-Labor, Technische Universität Berlin
Ernst-Reuter-Platz 7
D-10587 Berlin
{andreas.lommatzsch|thomas.strecker|danuta.ploch|
conny.strecker|robert.wetzker}@dai-labor.de

Abstract: In this paper we introduce a version of the Spree expert finding framework [BAA+07] tailored for enterprises. Whereas expert finding services have been very successful on the Web, enterprise level solutions are still scarce. This comes as a surprise, as the process of finding the right person (to ask) among colleagues requires a considerable percentage of most employees' time yielding a high potential for optimization. The core of Spree is an expert finding algorithm that automatically maps questions to the most qualified experts using a domain-specific topic taxonomy as intermediate. Apart from the framework itself, we describe the challenges and design decisions that have to be taken into consideration when implementing expert finding solutions in enterprises. These include the selection of an appropriate domain taxonomy, the motivation of employees to share their knowledge and privacy related concerns.

1 Motivation

The *classical* Internet provides information in the form of documents. Users searching for information can either directly access these documents or use search engines to identify the most relevant ones for a given query. However, documents contain only a fraction of the entire knowledge. In some cases, instead of finding a document, the user might want to find the right person to ask. Whereas the need for expert finding solutions has originated

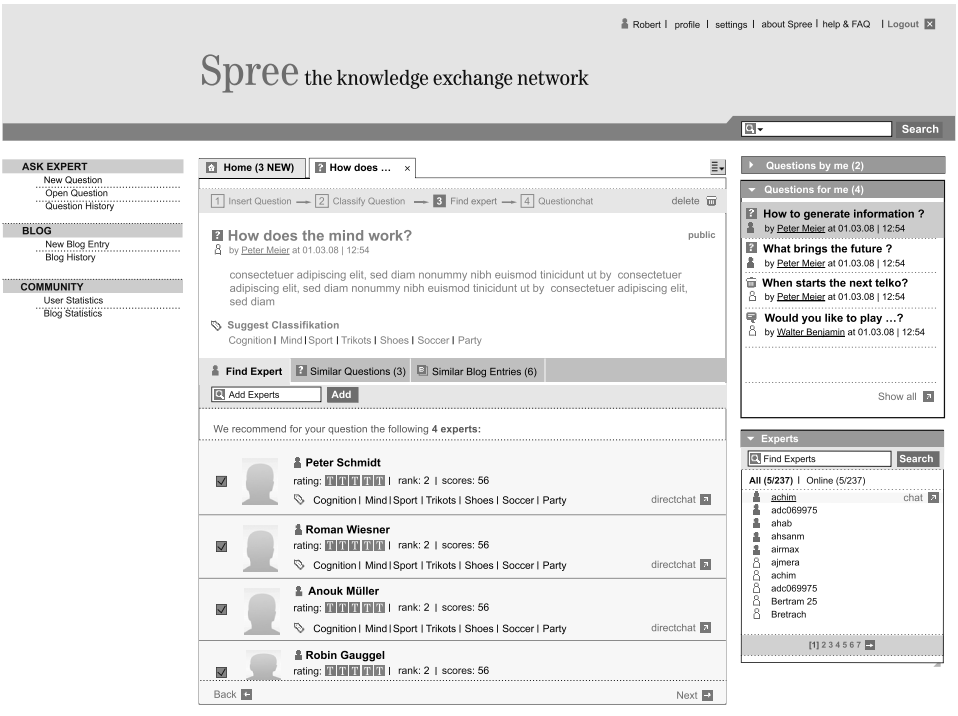

Figure 1: The Spree user interface.

various services on the Web, such as Yahoo! Answers[1] or MSN QnA[2], enterprise oriented solutions are still scarce. This comes as a surprise, as the process of finding the right person (to ask) among colleagues requires a considerable percentage of most employees' time yielding a high potential for optimization. This is especially true for larger enterprises where employees only know a small fraction of the entirety.

In this paper, we present the Spree expert finding tool [BAA+07] in a version especially tailored for enterprises. Spree is a web-based community platform that combines the search for experts with other technologies, such as blog and chat, to support the knowledge diffusion process among the employees of an enterprise and make human-centric knowledge explicit.[3] The core of Spree is the matching algorithm. The Spree system calculates the domain a given natural-language question belongs to and automatically identifies the experts among its users most qualified to respond. During this process we use topic taxonomies as intermediates and map questions and experts into the taxonomy space for efficient similarity calculations. The sound design of a domain taxonomy is crucial, as it directly accounts for the matching quality. The classification of text using hierarchical structures has been extensively studied in literature. It was shown that hierarchical struc-

[1] http://answers.yahoo.com/

[2] http://qna.live.com/

[3] The original Spree prototype focusing on communities is publicly available at http://www.askspree.de.

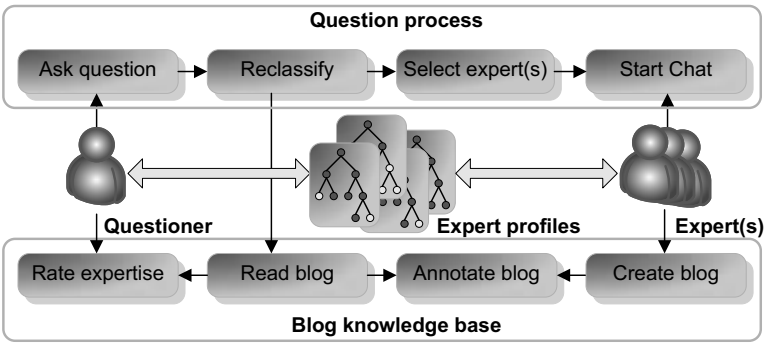

Figure 2: The *question* and *blog* process flows.

tures improve classification quality compared to flat approaches (e.g. [KS97], [DC00]).

The current implementation of the matching logic does not support typed categories, such as categories for locations in contrast to skill categories, or the assignment of additional category attributes. However, in future versions of Spree, we plan to integrate these concepts well known from ontologies in order to allow for a more complex knowledge design. In this paper, we will therefore not distinguish between the words *taxonomy* and *ontology*.

Experts are informed whenever a new question within their knowledge domain appears and can answer a question in real-time using the provided chat or email functionality. Additionally, users of Spree are encouraged to create blog entries about topics they consider interesting for other community members. These entries together with all questions and answers can be searched and rated by the community increasing the effectiveness of the framework over time. Figure 2 depicts the knowledge generation process.

This paper is structured as follows: Section 2 describes the Spree framework and its core components. We then outline the challenges related to the design of an enterprise knowledge taxonomy and present the ontology editor tool that helps during the taxonomy construction process. Section 4 presents the expert finding algorithm, the core functionality of the Spree framework. We then discuss the challenges we met during the development of Spree and conclude with an outlook on possible future improvements.

2 The Spree framework

For the development of Spree, a modular approach was chosen. This modularity allows the reuse of components, such that e.g. the expert finding functionality can be integrated into other possibly existing solutions. The Spree framework consists of six functional components: *User Management*, *Ontology Management*, *Request Management*, *Matching*, *Communication* and *Community*. We describe the Spree components and their interaction on the example of the *ask process* which includes the classification of questions as well as the expert finding and communication (Figure 3).

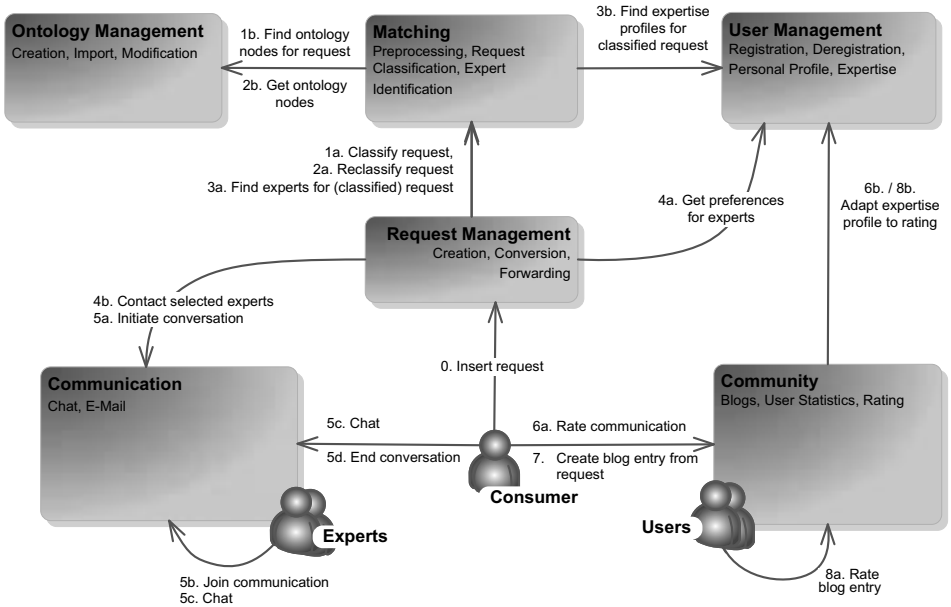

Figure 3: Active components within the *ask process* flow.

The central component within the *ask process* is the *Request Management* that manages the entire life cycle of a question including the creation of the corresponding request objects and its forwarding to the most appropriate experts. The component also handles user actions referring to requests, such as view, delete or close actions.

In order to find the best matching experts, the request has to be classified (step 1). Both the classification and the subsequent expert matching are handled by the *Matching* component. This component performs a language-dependent analysis of the request's textual representation and classifies the request to ontology nodes. The resulting classification is then presented to the user who can modify it if required. For the expert matching process, the matching component also interacts with the *Ontology Management* that provides the current ontology and ontology related functionalities, as well as with the *User Management* where user expertise profiles, user preferences and previously received ratings are stored.

The resulting expert list is then presented to the questioner who can add or remove experts. This modification step may be required in cases where the questioner already knows about potential expert candidates or wants to exclude certain experts from the process. The *Request Management* then forwards the request to the selected experts (step 4). If an expert is currently on-line, he will be informed in real-time about the incoming question. However, experts do not always have to be logged in but may also be informed about new requests via email if specified in their user preferences. The *Request Management* then triggers the initialization of a conversation between questioners and experts via the *Communication* component (step 5). Experts can reply to a request by sending an email

that will be forwarded to the questioner or, in the majority of cases, by joining a chat started for each request.

After a conversation has been finished, the consumer can rate it (step 6a) and all participants are asked to summarize the conversation in a blog entry (step 7) that may be found by potential future questioners. All rating and blog functionality is handled by the Community component. In addition to conversations, also blogs (step 8a) can be rated. An expert can thus improve his score by providing quality answers or by writing blog entries and augmenting the Spree knowledge base. The *Community* component is also responsible for updating the expertise profiles based on received ratings (steps 6b, 8b). Furthermore, the community component provides all functionality for presenting system statistics, e.g. high score lists or an overview about the most popular topics. The statistical data is designed to motivate community members to share their knowledge.

3 Taxonomy design

Designing a descriptive taxonomy is crucial for the success of the expert finding solution. Design decisions do not only include the specification of relevant categories and their arrangement into a hierarchical structure, but also require thoughts on the type of classification a later classifier should return and on the optimal taxonomy size.

For the category selection, it is crucial to define what expertise domains are essential. Here, expertise may be seen from a skill perspective where employees are considered experts based on their knowledge about technologies, products, business partners or markets. However, in some cases the taxonomy may simple reflect an enterprise's internal structure considering users as experts if they are responsible for a certain task or domain. The Spree approach allows unifying the skill perspective and a more structural view into a single taxonomy. This is made possible by the fact that Spree uses a multi-class classifier that categorizes texts to multiple branches of the taxonomy.

The selection of an optimal taxonomy size is also crucial. Even though the granularity of a large-scale taxonomy may cover all topics relevant for a given domain, the size of such a taxonomy remains a considerable obstacle both for classification and usability. On the other hand, small taxonomies are easier to design, result in better usability, but may not provide enough granularity to fully describe a domain. The number of categories should generally increase with the number of experts in the system. To solve this problem, Spree includes an ontology editor tool for the manual tailoring of taxonomies during run-time. An automated approach to the problem of taxonomy tailoring is described in [WUH+08].

Using the ontology editor, each topic node can be defined by a name and a set of meta-data, such as a description and a set of characteristic keywords. To each category, the editor can then assign documents that he considers descriptive. These documents are converted to plain text and analyzed based on natural language processing tools (see section 4.1). In a next step, the ontology editor creates an n-gram frequency statistic that will later be used to train the Spree text classifier. The editor also supports the modification of the taxonomy structure itself by adding new or removing existing categories. Furthermore, it is possible

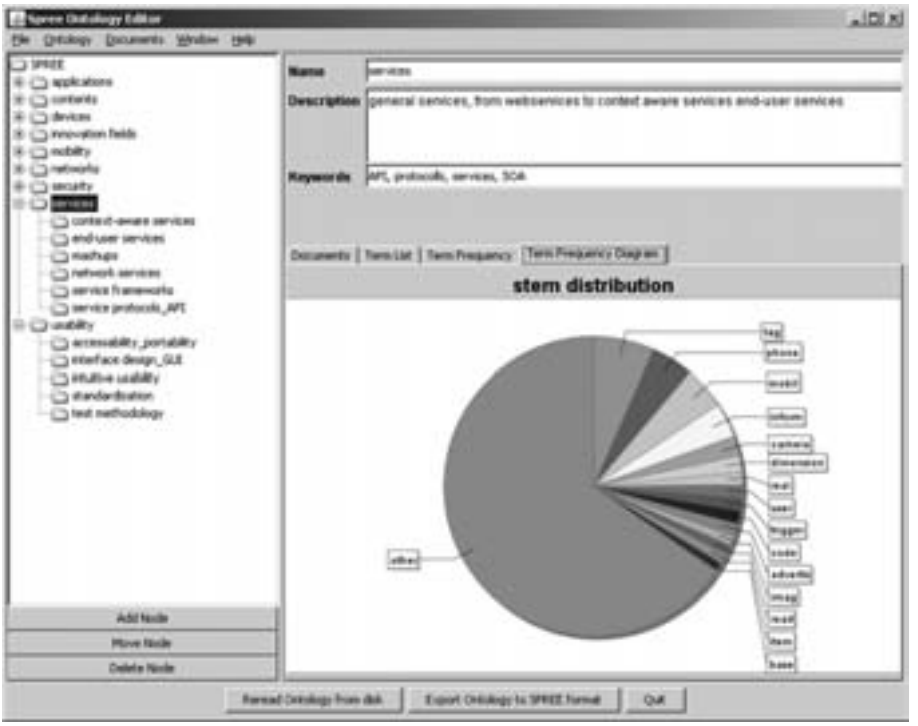

Figure 4: Screenshot of the Spree Ontology Editor shows the topic hierarchy (on the left side), the meta data and the visualization of the term statistic for the selected node.

to move categories or entire branches for more complex structural changes. Taxonomies designed with the ontology editor can be exported to a new Spree instance or to a file archive. The tool also allows making changes to the taxonomy of a running system which may be a requirement for dynamic domains.

The ontology editor interface is shown in figure 4. An enhancement planned for the near future is the automated suggestion of subcategories for topics, e.g. by using a cluster algorithm such as the one described in [Bis06] to group the documents assigned to a topic.

4 Finding the right experts

One of the core functionalities of the Spree framework is its matching algorithm. This algorithm classifies a given question to the topics of a predefined taxonomy and identifies the most qualified experts based on their expert profiles within the same ontology. Once the taxonomy has been designed and relevant documents and keywords have been assigned using the ontology editor, the Spree text classifier is trained. This classifier allows us to map a given question or document to the most relevant categories. Based on these

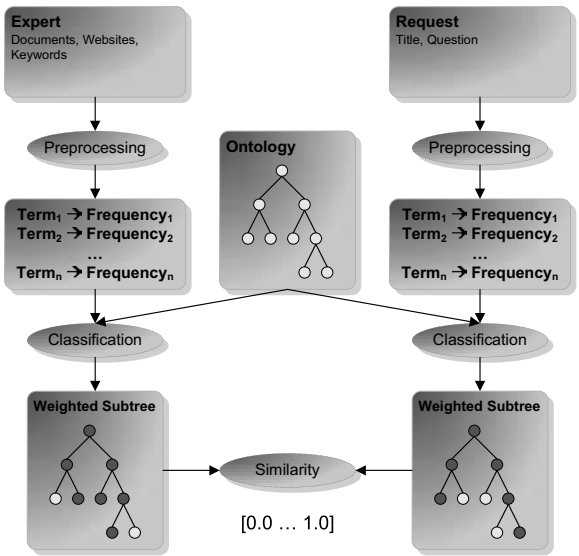

Figure 5: The Spree expert matching process that calculates the similarity of questions and experts using an ontology as intermediate.

categories, we can then identify the most qualified experts.

Figure 5 gives an overview of the Spree expert matching process which we will describe in detail in the consequent sections.

4.1 Text processing

The text processing task for the transformation of documents and questions is subdivided into multiple steps. First, a text parser splits the text input into tokens. The sorted token list is then used to extract the text n-grams where the parameter *n* can be set in the system parameters. Next, all n-grams that consist of stop words are removed from the n-gram set as they transport little information. The remaining n-grams are stemmed using the snowball stemming algorithm[4] and aggregated to n-gram histograms that are considered the internal representation of an input text. The text processing algorithm is applied identically to documents assigned to categories in the ontology editor as well as to user questions. The current implementation of the algorithm supports German and English texts.

[4] http://snowball.tartarus.org/

4.2 Text classification

The ontology editor assigns keywords and descriptive documents to each node of the ontology. These keywords and documents are transformed to n-gram distributions during the text processing step such that each topic node can be represented by a characteristic n-gram distribution (vector). As the provision of sufficient training documents and keywords may require a very high manual effort, it is also possible to learn characteristic n-gram patterns by using external sources such as Intranet or Internet search engines [WAB+07]. However, this automatic retrieval process may not produce satisfying results in domain specific scenarios with their own vocabulary.

The Spree framework is designed to support a variety of classifier types. The selection of a classifier type may depend on the number of available training material or the required classification speed. By default Spree will use a Naïve Bayes text classifier. For each node, this classifier estimates the likelihood that the corresponding n-gram distribution generated the n-gram sequence observed in the input text. The *m* most likely nodes are then considered valid classifications where *m* is a system parameter. Classifications are always complete in the sense that if a category is assigned to a given text also all parent categories are considered valid classifications. Classifications, therefore, always appear as subtree of the taxonomy.

4.3 Expert matching

The Spree expert matching algorithm identifies experts to a given question based on the ontology tree T whose Nodes $N = n_1, \ldots, n_N$ correspond to the different knowledge areas. Apart from the structural assumption, the Spree system remains independent of the nature and content of the ontology considered in any implementation. The fundamental idea of the matching algorithm is to represent experts and user questions as serialized vectors of nodes $v(T) \in S(T)$ where $S(T) \subset R_N$ is the ontology space. The values of v(T) are set to 0 or 1. Once all registered experts $e_1, ..., e_E$ and an incoming question q have been mapped to subtrees, it is possible to compute the similarity between an expert and the question by calculating the weighted dot product

$$score(q, e_i) = v(q) W v(e_i)$$

where W represents a weight matrix that allows us to incorporate further contextual information about the topics of the underlying ontology. A question is then forwarded to the experts with the highest score.

A detailed description of the Spree matching algorithm is given in [BAA+07].

5 Discussion of practical experiences

Whenever a new application or tool is introduced in a large enterprise, several aspects have to be taken into consideration. Most time consuming, at least in Germany, is getting the agreement of the workers' council, as any handling of personal data requires its agreement. To convince all involved parties, clear concepts are needed from the very beginning covering the following items:

- Data protection concept to fulfill legal issues
- System security concept to meet basic IT requirements
- Operational concept to run the application
- Business plan to convince the upper management
- User guidelines to finally reach the people

Community tools on the Web work because of the voluntary contribution of their users. For enterprises, people have to see the advantages of their participation, too. As known from the e-learning and knowledge domain, even though incentives are often under discussion, but public scores that help to motivate users and increase trust in most public scenarios are often rejected as devil's work. The advantages and disadvantages of expert scores have to be evaluated for each possible domain separately and then communicated to the management. Despite these incentives, general software quality requirements are essential for success. Especially usability aspects are of crucial importance. Users have to easily understand the application's scope and functionality. This requires the application design to follow the design guidelines also found in existing applications as well as a possible integration into existing solutions. Compliance requirements generally include:

- Compliance with the standard IT workplace: hardware, software versions, security and browser settings, user execution rights, etc.
- Compliance with the IT production platform: virtual servers, database environments, data protection guidelines, etc.
- Compliance with the enterprise user administration: LDAP authentication, single-sign-on, password guidelines, etc.
- Compliance with corporate design: (Web-)design guidelines, possible integration into existing portals, etc.

User satisfaction also depends on the quality of answers and time for receiving a response. The quality of the expert profiles and the underlying matching algorithm is therefore crucial. Also, to overcome the well-known cold start problem of community tools, it is necessary to motivate a sufficient participation rate especially in the early phase.

Spree can be seen as a flexible process overlay, not as a replacement. Therefore, interest conflicts can occur, giving the management the feeling that working time is stolen. However, the savings are much bigger by avoiding double work, and the staff satisfaction index will increase. Keep always in mind, the knowledge of people is the most valuable thing a company has.

6 Outlook

There exist a variety of possible directions for further improvements of the Spree enterprise solution:

Automatic profile creation: There exist many sources for the automatic creation of employee expertise profiles. One very promising source are the documents an employee has created or worked with. The topics of these documents could be used to estimate an initial expertise profile. This idea is especially appealing as enterprises generally possess a central file storage system. For an example on automatic expert finding based on documents see [SHFA07].

Dynamic profile updates: If a user successfully answers questions in areas not listed in his expert profile, he should be asked to update his profile. The same should happen if he has authored documents related to other topics.

Learning classifier: Users give feedback when they accept or modify a proposed classification during the ask process or the creation of a blog entry. This feedback is a valuable source for a continuous improvement of the classification accuracy.

Visualizing the enterprise knowledge graph: Tools that help to visualize the knowledge structure and diffusion processes within an enterprise provide an invaluable source for management decisions. The right visualization may also guide users during the process of expert(-ise) finding, as done e.g by the SkillMap tool[5], and emphasize areas where an enterprise lacks domain knowledge or skills. The community version of Spree already provided a visualization tool, the Spreegraph[6], for browsing the social and knowledge graph. For future work, we plan to extend this existing solution toward the Enterprise 2.0 scenarios.

Limit effects of malicious behavior: Wherever users can rate other users, there exists an incentive to cheat. The effect of cheating should be reduced, e.g. by allowing users to rate other experts only once.

Integration with existing Document Management Solutions: To allow questions and discussions about certain documents, there should be a tight binding between Spree and existing document management solutions (DMS).

[5] http://ioe-skillmap.hu-berlin.de/

[6] http://www.askspree.de/static/flash/SpreeMainComponent.html

References

[BAA+07] Christian Bauckhage, Tansu Alpcan, Sachin Agarwal, Florian Metze, Robert Wetzker, Milena Ilic, and Sahin Albayrak. An Intelligent Knowledge Sharing System for Web Communities. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics 2007*. IEEE Computer Society Press, 2007.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[DC00] Susan Dumais and Hao Chen. Hierarchical classification of Web content. In *SIGIR '00: Proc. of the 23rd annual int. ACM SIGIR conf. on Research and development in information retrieval*, pages 256–263, New York, NY, USA, 2000. ACM Press.

[KS97] Daphne Koller and Mehran Sahami. Hierarchically Classifying Documents Using Very Few Words. In *ICML '97: Proc. of the 14th Int. Conf. on Machine Learning*, pages 170–178, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[SHFA07] Pavel Serdyukov, Djoerd Hiemstra, Maarten Fokkinga, and Peter M. G. Apers. Generative modeling of persons and documents for expert search. In *SIGIR '07: Proc. 30th int. ACM SIGIR conf. on Research and development in information retrieval*, pages 827–828, New York, NY, USA, 2007. ACM.

[WAB+07] Robert Wetzker, Tansu Alpcan, Christian Bauckhage, Winfried Umbrath, and Sahin Albayrak. An unsupervised hierarchical approach to document categorization. In *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 482–486, Washington, DC, USA, 2007. IEEE Computer Society.

[WUH+08] Robert Wetzker, Winfried Umbrath, Leonhard Hennig, Christian Bauckhage, Tansu Alpcan, and Florian Metze. Tailoring Taxonomies for Efficient Text Categorization and Expert Finding. In *WI '08: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, Sydney, Australia, 2008. IEEE Computer Society.