

Matching Human Actors based on their Texts: Design and Evaluation of an Instance of the *ExpertFinding* Framework

Tim Reichling
University of Siegen
Hoelderlinstr. 3
57068 Siegen
+49 271 740 3383

reichling@fb5.uni-siegen.de

Kai Schubert
University of Siegen
Hoelderlinstr. 3
57068 Siegen
+49 271 740 3383

kai.schubert@uni-siegen.de

Volker Wulf
University of Siegen
Hoelderlinstr. 3
57068 Siegen
+49 271 740 2910

wulf@fb5.uni-siegen.de

ABSTRACT

Bringing together human actors with similar interests, skills or expertise is a major challenge in community-based knowledge management. We believe that writing or reading textual documents can be an indicator for a human actor's interests, skills or expertise. In this paper, we describe an approach of matching human actors based on the similarity of text collections that can be attributed to them. By integrating standard methods of text analysis, we extract and match user profiles based on a large collection of documents. We present an instance of the *ExpertFinder Framework* which measures the similarity of these profiles by means of the Latent Semantic Indexing (LSI) algorithm. The quality of the algorithmic approach was evaluated by comparing its results with judgments of different human actors.

Categories and Subject Descriptors

H.3.1, H.3.3

General Terms

Algorithms, Measurement, Experimentation, Human Factors, Verification

Keywords

Expertise Sharing, Latent Semantic Indexing, Keyword Extraction, User Profiling, Community Building, Knowledge Management

1. INTRODUCTION

Terms like *social* or *human capital* have gained remarkable meaning during the past few years. These terms refer to human resources like abilities, social networks or explicit and implicit knowledge of employees in organizations. Cohen and Prusak [9] predict that there is a high potential for companies to increase productivity and speed of innovation cycles by enabling and

fostering these resources. Accordingly, *knowledge management* (KM) technology and strategies aim at utilising these resources within organizations or workgroups. Unfortunately, to date, no generally applicable (and easy) way to utilize these resources has been found so far.

The term *knowledge*, as we know it, covers two distinct forms. *Explicit* knowledge that can easily be externalized to documents or certain binary forms, and *implicit* or *tacit* knowledge which is given by the experience or practices is closely bound to human actors [11]. As a result, KM systems focussing purely on explicit knowledge (repository-based approach) are highly dependant on the existence of explicated knowledge and thus might be very limited in what they can offer to users.

Bridging this gap and making implicit knowledge more accessible has become a key issue in the recent KM debate. Ackerman [2] and Ackerman and McDonald [1] in describing the concept of the 'Answer Garden' (and later Answer Garden 2) have made an early attempt to link repository-based approaches to KM with those based on human expertise to create a 'vital' and dynamically growing knowledge repository. The application of the Answer Garden approach requires, however, specific preconditions which are not always met by potential fields of application [18].

Expertise recommender systems were developed in order to derive social and human capital as opposed to content that is more static in nature. Examples of those systems are *Expert Finder* [29] or *Expertise Recommender* [16] [17]. In trying to support social capital, Becks, Reichling and Wulf [7] describe an early instance of the *ExpertFinding Framework* (see below). The system was applied within an e-learning platform to closer connect learners to each other based on common interests and synchronously accomplished training units.

All of these systems create and store user profiles by interpreting structured data, such as histories of interaction, Java source code, standardized comments within source code, classification schemes within repositories, etc. which can be attributed to human actors. The algorithmic interpretation of these artefacts with regard to a human actor's interests, skills or expertise is anyway a difficult and a risky undertaking from a privacy- or security standpoint, and gets even more difficult in cases where personal unstructured data (for instance, arbitrary texts, foreign languages, etc.) is concerned.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GROUP '05, November 6–9, 2005, Sanibel Island, Florida, USA.

Copyright 2005 ACM 1-59593-223-2/05/0011...\$5.00.

It is also interesting to notice that a proper evaluation of the quality of the recommendations generated by these systems is often lacking. Obviously, the quality of recommendations depends on (1) the reliability of the personal data (2) the accuracy of the derived user profiles and (3) the performance of the matching algorithms taken to compare those profiles.

In this paper, we present an instance of the *ExpertFinding Framework* which generates profiles and matches users based on collections of unstructured texts. The prototypical system is called TABUMA which stands for *text analysis based user matching algorithm*. We also did an empirical evaluation on the quality of the matching recommendations generated by the framework.

2. PREVIOUS WORK

In this section, we give a detailed overview on previous approaches and systems in the field of KM that are closely related to our work. To ground our implementation, we also present methods and concepts taken from the fields of text analysis and information retrieval.

Note: When reading the following section, be aware of several systems and approaches using similar names which may lead to some confusion.

2.1 Recommender systems

The term *Recommender system* (RS) was initially introduced by Resnick and Varian [21]. In accordance with Runte [22], we roughly distinguish between non-personalized and personalized systems. While non-personalized RS generate recommendations independent of the requesting user, personalized RS strongly take certain users preferences and attributes into account and thus generate individual results for each user. The theoretical approach of recommending is fairly simple: Users are offered ways to specify a request. In general, this can be done by entering keywords or giving a natural language description of the problem. As a result, the RS (in general) returns contents concerned with the specified problem. Since such results are required to exist in externalized form this approach is not capable of covering implicit knowledge. Alternatively systems may return references to human actors that are identified as ‘experts’ in the requested domain. The latter can be seen as an attempt to offer access to implicit knowledge that cannot be externalized. We will refer to the latter approach as *Expertise Recommender Systems* (ERS). To point out the character of ‘going beyond explicit knowledge’ the term *Extended Recommender System* would also make sense.

When regarding approaches out of the research three types of data appear to be most important in terms of generating user profiles:

1. *History data*: Bookmarks or browser histories are user historical data sources that are likely to represent the user’s interests or competencies.
2. *Documents*: Documents that were written, read or reviewed by a user appear to be another indicative source of data for user assessment. ‘Documents’ include text documents as well as emails or newsgroups postings.
3. *Social network*: Analyzing the user’s social environment is another method of user assessment. The

basic idea is that users that have collaborated in the past are likely to collaborate successfully in the future.

We will now have a closer look at some certain approaches and systems taken out of a large set of systems. Since it is not possible to give a complete overview of all the different systems and approaches we try to give a closer look at a representative choice of ERS (see also [16]).

The system *Siteseer* [23] utilizes the user’s web bookmarks to assess his or her personal interests and thus can be seen as belonging to the first category of ERS. Based on the user assessments Siteseer learns to categorize websites by means of common attributes of the sites visitors. This way Siteseer offers personalized recommendations for websites. In contrast, the system *PHOAKS* (*People Helping One Another Know Stuff*) [27] and *GroupLens* [20] utilize newsgroup-postings to generate appropriate user profiles.

HALe [14] analyzes the users’ emails to learn about relationships between users among each other. Additionally, relationships between the users and ‘issues of interest’ are extracted from the mail texts. Both are done using Part-of-Speech (POS) Tagging and other linguistic methods including latent semantic indexing (LSI). A similar approach of modelling the actors’ social networks based on analyzing the email traffic is presented by Tyler et al. [28]. This approach focused on discovering Communities of Practice (CoPs) within organizations that coexist with the formal organizational structure in a subtle way. Members of CoPs in this approach are identified by a strong mutual mail exchange (exceeding a certain minimum threshold). The ERS *Fab* [5] was developed by the University of Stanford. Fab combines content- and community-based recommendations. The user assessment is done by analyzing texts that were commented or evaluated by the users.

Based upon the finding that social relationships are a key factor for cooperation within organizations the *Expertise Recommender* system [15] was developed. The system supports filtering its recommendations according to the requestor’s social network which is modelled within the system in two different ways: The *Work Group Graph* (WGG) reflects the formal organizational structure relying on persons within the same department to have a closer relation to each other than those of different departments. Thus the realized matching method is called *Departmental Matching*. In contrast *Social Network Matching* relies on a model of the (real) social relationships that is gathered by ethnographic methods like interviews, observations or evaluation of certain users’ artefacts – which in general is highly expensive.

Referral Web [13] is another approach aiming at utilizing the actor’s social network to give recommendations. The goal is to recognize as well as create communities based on co-authorship in scientific papers. Recommendations are driven by means of social relations. In more detail a chain of persons is created such that two successive persons within the chain are known to each other. The shorter the resulting chain is ‘closer to each other’ should the persons (represented by the end points of the chain) feel to be.

An ERS playing a major role for the system presented here is given by the *ExpertFinding Framework* [7]. The first prototype of this framework was developed as component of an e-learning

platform, which has proved to be promising (see [7]). Learners within the platform were assessed according to their learning history. By having well structured training units dedicated to well-defined domains generating the user profiles could easily be done. Furthermore, as the material was learning stuff, consuming such material can reliably be interpreted as an indication of interest in a certain domain. Additionally, as the registration required the users to give some information about their education and apprenticeship as well as their carrier path this information was another good indicator of knowledge, interests or skills.

Based on the existing information about the users two matching strategies were implemented within the ExpertFinding Framework: *Profile Matching* takes the education and experience of users into accounts to compute the ‘fit’ between two users or a user and a request respectively. Based on this value a recommendation is returned. The same is done by *History Matching*: Instead of education and experience the learning history is taken to generate the fit. Both methods could be ‘mixed’ according to the requestors preferences: If the requestor was interested in a high degree of expertise within certain professions Profile Matching will be a good choice as the career path here can be seen as highly indicative. If on the other hand, the requestor was interested in co-learners being involved in the same kind of learning stuff – and thus were not far away from oneself – History Matching should be the better choice. We will pick up the ExpertFinding Framework again in section 3 as it will be the fundamental concept for the prototypical system presented in this paper.

While most of the existing systems described above use structured documents or data (e.g. bookmarks), our approach differs in that it uses unstructured text files. In most cases, the amount of unstructured documents created by a user very often overtakes the amount of the structured ones. To handle this type of data, we need methods of automated text analysis.

2.2 Methods of automated text analysis

Since text files are the main source for user profiling in our approach, we have to deal with automated text analysis in order to generate the profiles. Several methods of algorithmic text analysis are known so far. Many of them can be assigned to the research field of *Text Mining*. Identifying key terms and major concepts as well as recognizing most important fragments from unstructured text are major goals of Text Mining [26].

In the following, we give an overview on the different methods of statistics, neural networks, the *Vector Space Model (VSM)* and *Latent Semantic Indexing (LSI)* and heuristics. A comparison of advantages and disadvantages of these methods completes the subsection. Interesting methods of textual analysis can be taken from the field of linguistics. While *morphology* aims at text segmentation into morphemes (smallest meaningful units), the goal of *syntactical analysis* is to describe natural language by a formal grammar. Even though this can easily be done with programming languages it is not applicable for native speech as it is used in spontaneous speech, emails or notices. However, as we argue in the end of this section for several reasons linguistic methods appear not to be appropriate for our goals.

Statistical methods

Statistical methods can be seen as the earliest methods of textual analysis that ever were applied. Subject of these methods are terms that are examined by means of their absolute or relative

frequencies within texts or sets of text. In order to identify key terms within a given text [12] uses ‘common’ reference texts and distinguishes four categories of terms:

- 1) Terms that appear within a given text but do not appear within the reference text are highly relevant key terms
- 2) Terms appearing in a given text with a higher relative frequency than in the reference text are likely to be key terms
- 3) Terms appearing in both text with a similar relative frequency are not likely to be key terms
- 4) Terms appearing in a given text with a lower relative frequency than in the reference text are not likely to be key terms

Besides these definitions of identifying key terms within text other useful statistical methods can be used alternatively or in addition to complete the key term extraction: *Markov-models* offer a model of morphemes capable of predicting morpheme chains using statistically identified values of transition probabilities.

Collocations describe tuples of terms (generally pairs or triples) that appear to have a ‘common meaning’. Indicators of collocations – in contrast to arbitrarily chosen tuples of terms – are statistical and semantic evidence of mutual meaning. Accordingly, collocations require significantly frequent co-occurrence of terms within sentences as well as a common semantic context they belong to. Examples of collocations might be “grass – green” or “sky – blue”. Collocations can efficiently be identified within large text collections using methods of data mining. However, for the purpose of identifying keywords (that can be used like single terms) we will focus on collocations of consecutively occurring words. Additionally, those ‘simple collocations’ are much easier to detect (see below).

Stemming

A reduction to the principal forms can be carried out to identify terms that are a member of the same family (for instance “play”, “played” and “plays”). Indexing each version appears not to be useful regarding memory requirements and runtime issues. Three different methods of reduction are taken into account here:

- 1.) A very simple method is called *trunking*. It simply ‘erases’ a defined number of letters from the end of a word. Trunking is not applicable for several languages, including German.
- 2.) *Stemming* is a method that reduces the lengths of words by the use of sophisticated algorithms depending on the language used. Hence for each language a dedicated implementation is needed. An algorithm for English has been developed by Porter [19], and a modification of this algorithm for German is given by the *snowball German stemming algorithm* [24].
- 3.) *Dictionary-based* methods are delivering the best results (depending on the quality of the dictionary used). Since dictionaries can become incredibly large and the method is highly time-consuming these methods are not applicable for our purposes (see below).

The Vector Space Model (VSM) and Latent Semantic Indexing (LSI)

In order to represent and compare text documents¹ in an appropriate way we focus on two further methods, the *Vector Space Model* (VSM) and *Latent Semantic Indexing* (LSI) that builds on the VSM. The basic idea is to represent documents by vectors in a multidimensional ‘semantic space’ that is given by the overall set of terms that occur within all the regarded documents. Each of the documents is transformed into a vector within that semantic space: Vector components are given by the according term frequencies in the regarded document. In general, this transformation is carried out after filtering out stop words (see below) and extracting key terms in order to decrease the overall amount of terms. The VSM gives us a good base for representing, storing and comparing documents since well defined metrics, distances and other vector space operations are inherited. For instance, in order to measure similarity (or rather distance – which can be interpreted as ‘non-similarity’) between two documents the Euclidean metric can be used. Alternatively, as depicted in figure 1 for the case of two dimensions, using the dot product, the ‘angle’ between two documents can be computed as well [26].

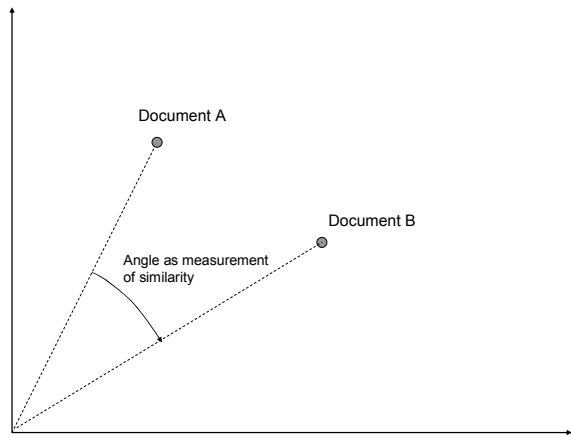


Figure 1: Documents represented as Vectors in the VSM

Given a set of n documents containing an overall amount of m (different) terms leads to n (row-) vectors of the size m which can be interpreted as an $m \times n$ -matrix – the *term-document-matrix*. Since each of the documents is likely to contain only a small subset of the overall set of terms it turns out that the term-document-matrix for a given set of documents (assuming that n is clearly greater than 1) is very sparse – mainly consisting of zero values. The question arises of whether this matrix can be compressed such that the same information can be represented more efficient. Another question arising is, how to handle synonymy and ambiguity among terms that typically comes up when documents from different contexts are regarded.

A very promising approach of handling these problems is given by *Latent Semantic Indexing* (LSI) [10]. As a detailed description of this method would exceed the focus of this paper we are

¹ The term ‘document’ is a substitute for any kind of data set consisting of text. Thus user profiles consisting of keyword listings are referred to as well.

restricted here to mention some basic properties of this method: LSI is built up upon the VSM. The term-document-matrix is assumed to be given. By utilizing a method taken from linear algebra the so called *Singular Value Decomposition* (SVD), the multidimensional semantic space represented by a given term-document-matrix (A) can be compressed by means of reducing its dimension to an arbitrarily chosen degree k with $1 \leq k \leq n$. In the resulting k -dimensional vector space (represented by a compressed $k \times n$ -matrix A’) the documents are ‘forced to move closer to each other’ – since $k \leq n$. It appears that those documents having a ‘latent similarity’ to each other now tend to create groups within the shrunk space. This latent similarity can be expressed by the angle between documents. It further appears that problems coming up by means of synonymy can now be handled as documents located within the same domain – and thus containing terms with similar meaning – are located close to each other in the resulting semantic space (whereas ambiguity is still a problem). The choice of k influences the accuracy of the result in the following way (as Berry, Dumais and O’Biran [8] state): “While a reduction in k can remove much of the noise, keeping too few dimensions or factors may loose important information”. So k should be chosen carefully according to the defined goals: A small value of k reflects the origin semantic space in a coarse way, revealing very subtle relations between documents, while a large value gives an accurate impression that is closer to the original. For further details about the LSI method we refer to Deerwester et al. [10].

Further methods

Aside from the methods described so far other methods or even domains appear to be interesting at first view. However, certain disadvantages let these methods appear inadequate in terms of the goals we aim at. First, the use of neural networks seems to be an interesting option. Because of the huge effort to ‘train’ these networks (and some other special characteristics), they are inapplicable for our needs. Another option can be seen in heuristical methods (e.g. ontologies or the evaluation of textual structures). Many documents types contain structure tags or meta data (e.g. the subject field in emails or headings in texts). Depending on the field, the abundance or lack of standardized formats and the generally poor support actually inhibits its use.

We realize that a combination of different methods of text analysis appear to be promising in that we can automatically analyze text documents, create user profiles, and match these with each other. However, in our current approach we use statistical methods which seem to offer good way of keyword extraction from text documents that are used to create the user profiles. Since, in general, large amounts of texts have to be analyzed, we expect the statistical methods to outperform alternative methods in both run time and quality of results. In order to compare the profiles in the next step LSI seems to be a good choice as problems of synonymy are widely avoided. The next section gives an overview of our approach.

3. THE TABUMA APPROACH

In this section, we describe the TABUMA approach. First the theoretical backdrop that we rely on is explained. Then we describe the *ExpertFinding* Framework that will be the technical basis in which the TABUMA prototype will be included. Finally the TABUMA approach is explained step by step.

Theoretical approach

Based on previous findings about the value of social and human capital we draw on fostering these concepts by enabling people to efficiently seek for other actors having certain attributes and get in contact with those persons. Since this is the first step to create a ‘work group’ or ‘community’ as enumerated by Wenger [30], the prototype itself does not assist the actors to actually create social capital. In fact, the actors themselves are required to build upon the system generated contact and may create a closer relation to this contact and thereby build social capital. As such, the role and support of the prototype is limited to some kind of ‘moderator’

The core questions here are (1) which are the properties of people that other actors seek for, (2) (how) can these properties be modeled in a binary form, and (3) (how) can these properties be gathered by accessing certain binary data sources in order to create user profiles. Further questions arise from this when thinking about privacy issues or personal motivation: Are actors willing to give a system access to their personal data? And will the actors cooperate in case of a recommender system working well? The latter two questions are certainly lying beyond the scope of this paper even though they address some very interesting and critical issues that would need further investigations.

Focusing on the questions (1) to (3) above, the *ExpertFinding* Framework aims at tackling these problems by offering a flexible architecture capable of including arbitrary data sources (see below). Assuming that abilities, interests and experiences in fact are matter of other actors interests (which still has to be further explored) we implemented and evaluated the TABUMA prototype in order to explore the capability of text documents to reflect the actual users interests, abilities and experiences (referring to questions (2) and (3)). Regarding the overall framework, we propose that several other data sources (like bookmarks, surfing histories, etc.) should be taken into account additionally in order to create more complete and indicative user profiles. However, in this paper text documents only are considered as data sources.

Architecture of the ExpertFinding Framework

We have embedded the TABUMA implementation within the architecture of the ExpertFinding Framework – thus enriching its profiling and matching functionality. Since this Framework is the technical basis for the implementation, it is briefly presented here. Technically, the approach includes two particular steps that are carried out asynchronously: (1) *Profile creation* from different data sources and (2) *matching* (comparing) the profiles with each other. Step 1 can be seen as an ongoing action that is peripherally performed in order to keep the user profiles up-to-date. Step 2 is carried out on a user’s demand.

The ExpertFinding Framework in its early implementation was embedded within an e-learning platform. The personal data derived from the e-learning platform were very specific. Accordingly, the same applies to the matching algorithms. When turning to new fields of application, we decided to develop the system to be much more flexible in the following ways: First, since potential sources of personal data cannot be foreseen when turning to a new field of application, the system is required to allow a rapid integration of (yet unknown) data sources. Second, since the actors preferences in terms of matching functionality cannot be foreseen as well, the system needs to allow a rapid

extension of the matching functionality. These two requirements have directed the systems design towards a flexible and modular architecture which is depicted in figure 2.

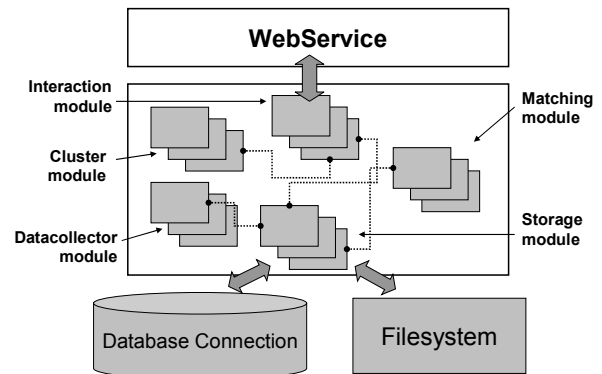


Figure 2: Architecture of the ExpertFinder Framework

The functionality offered by the ExpertFinding Framework is encapsulated within modules (depicted as small grey boxes) that can be developed independently while the frameworks main part is to create, run and occasionally stop and remove the instances of the modules. For the sake of brevity, we only describe those parts of the architecture that appear to be meaningful within this paper: First, *datacollector* modules are defined to access arbitrary sources of personal data. Second, in accordance with the type of data that is gathered by the datacollector modules *storage* modules persistently store this data. This can be done using the ordinary file system or a database. Third – again in accordance with the type of data that is stored – appropriate matching strategies can be implemented within *matching* modules. In cases where data sources cannot be directly accessed from the frameworks server machine ‘local agents’ are necessary, capable of gathering the requested data and transmitting it to the framework. This is likely the case for text documents or other artefacts that are commonly stored on the users’ local hard disk. In order to offer a standardized way of data exchange between the framework and (external) applications we rely on the *WebService* standard including technologies like XML and SOAP. This way inputs (i.e. for updating user profiles or requests) as well as outputs (i.e. the result lists or distance matrices) can be handled via standardized communication technology.

Text Analysis and Matching of Profiles

The implementation of the TABUMA approach was designed in order to carry out the two steps of profiling and matching as mentioned above. Figure 3 depicts the basic steps that are carried out in order to generate the user profiles from given text documents that are matched against each other on demand. We assume that the starting point of our approach is a given set of text files that is associated with the users work. We further assume that the files are already given in a plain text format rather than in a proprietary format – which is done in previous steps that are not part of our concept. However file filters are implemented, capable of extracting plain text from common text file formats (like MS Word, PowerPoint or PDF).

The first step is *Language recognition* which is necessary in order to carry out the following steps. Language recognition is very

similar to stop word filtering. It is performed by analyzing the known stop words of each (supported) language occurring within the documents. Since most texts can be assumed to be written using one main language, stop words from this language should occur much more frequent than those of other languages. Our implementation provides support of recognizing English and German – thus covering all the documents that were used within the evaluation (see below).

The next step *Stop Word Filtering* cannot be carried out without having determined the language. Since Stop Word Filtering is done by simply using stop word listings (one listing for each supported language) filtering stop words of the wrong language can cause filtering out meaningful words (for instance “die” in German is an article and thus a stop word whereas in English it is a meaningful term). In general, terms of the following categories can directly be assigned to be stop words: Articles, verbs, adjectives, adverbs, prepositions

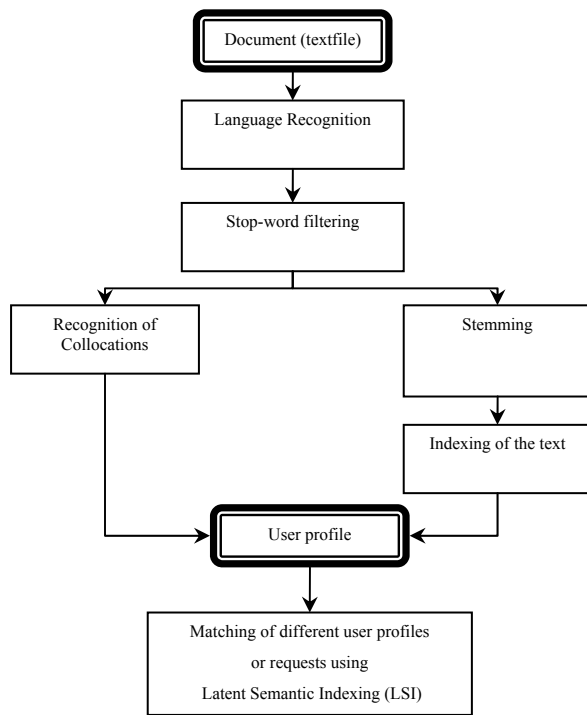


Figure 3: The TABUMA approach for automatic text analysis

The next steps are carried out in parallel threads as they cannot be done serially. First, we will describe the right side of the diagram in figure 3: In order to further reduce the overall amount of (different) terms, stemming is performed. Since the methods of stemming are highly dependent on the language this step cannot be carried out earlier. Furthermore, dependant on the language the quality of the results varies: While English texts were stemmed reliably by the Porter-Stemming algorithm [19] stemming German text using the *snowball* algorithm [24] provides ‘strange’ results in some cases, cutting away meaningful parts of words. Regarding these problems stemming has been implemented as an optional step. However, for the purposes of evaluation we have included stemming when conducting the tests. The results of the previous steps are then indexed and the term frequencies are

computed. As a result of these steps *frequency vectors* are returned to be joined together to a user profile.

In a parallel thread collocations are recognized. Since stemming may destroy meaningful collocations this step has to be carried out in parallel. The results of this recognition finally are returned in the same way as it is done with the ‘single’ keywords that are computed synchronously. Recognizing collocations (according to our restricted definition as two consecutive terms) is performed in a straight-forward manner by simply counting occurrences of consecutive pairs of terms within the text. In order to separate meaningful collocations from ‘ordinary’ pairs without a common meaning, certain criteria must be fixed to decide whether or not a given pair is to be classified as a meaningful collocation. For the TABUMA approach a fairly simple method to handle this has been chosen: By fixing a ratio of 0.1 between collocations and simple keywords it is guaranteed that 90% of the overall amount of keywords consists of single keywords whereas 10% are collocations. Even though more sophisticated methods of detecting meaningful collocation can be considered, for our purposes, this simple method has proven to return useful results (i.e. no meaningful keyword or collocation were left out). Since this problem is not part of the main focus of our investigations, no further effort has been done on this issue.

Both terms and collocations are joined together to form the final user profile in the next step. The final step of matching profiles is carried out on demand using the LSI method. The returned result – as mentioned above – is given as an ordered list (ranking) of user profiles (referring to persons) that best fit the request. In the further sections these listings will also be referred to as ‘recommendations’ or ‘assignments’.

To calibrate our approach, we had to find appropriate parameters at three points. First, we had to define the dimension of the profile vector. Second, we had to fix the ratio between keywords and collocations. Third, we had to determine the dimensions of the LSI’s semantic space.

To find these parameters, we conducted a preliminary study. Papers of nine students from different programs were asked to provide papers representing their current work. The set of papers contained 260.000 words in total. The amount of words was rather equally divided between the nine actors. We compared the results of different parameter settings with our judgement on the similarity of their programs and courses. By successive increasing the profile sizes we found out that no meaningful terms were left out when profiles reached the size of 500 weighted terms. Furthermore, the ratio of 0.1 between single key words and collocations turned out to be useful (see above).

Another observation that we encountered might sound a little surprising: When varying the LSI dimension it turned out that the best results were achieved with a dimension of 3 – which is a pretty small value compared to those that are used within the literature about the LSI method. Here values of ≥ 40 are recommended.

4. EVALUATION

In this section, we will first describe the methods we applied to evaluate our approach and then give an overview of the results we gained. Besides results concerning the quality of the recommendations some technical findings came up that appeared to be worth mentioning as well.

4.1 Methods

To evaluate the quality of recommendations made by the system, we drew on a network of researchers in applied computer science working at five different institutions (three universities and two research institutes). The five institutions are located in different cities up to 140 miles apart from each other. However, the research network has a common history as it grew over the past 10 years. Joint research projects are conducted. 18 researchers took part in the study. While they all knew each other, their history in the network and the resulting degree of mutual knowledge differed considerably.

All participants were asked to provide us with ‘typical’ text documents, taken ‘out of their work’. No restrictions in terms of the kind or the number of documents were made, except that the documents should be meaningful and indicative². As a result, the amount of documents given by each of the contributors varied highly: While some of them did not release more than one document others provided up to 26 documents. Table 1 gives an overview of the actors and their document selection. Mainly publications and working papers were provided in both German and English language. The system was fed with these documents to create matching recommendations for each of them.

Table 1: The network of researchers in more detail

| Person ID | Number of documents | Amount of text (given by the total number of words) ³ |
|-----------|---------------------|--|
| 1 | 6 | 22848 |
| 2 | 2 | 19667 |
| 3 | 1 | 8585 |
| 4 | 1 | 4602 |
| 5 | 1 | 7886 |
| 6 | 17 | 75568 |
| 7 | 9 | 50454 |
| 8 | 7 | 33954 |
| 9 | 2 | 22353 |
| 10 | 4 | 39269 |
| 11 | 2 | 9579 |
| 12 | 18 | 131512 |
| 13 | 6 | 20619 |
| 14 | 8 | 20730 |
| 15 | 3 | 16054 |
| 16 | 5 | 38680 |
| 17 | 1 | 7420 |
| 18 | 26 | 197852 |

² We preferred this way over simply utilizing each of the actors’ accessible publications to be sure that the documents really reflect the actual actors work.

³ One single page of ASCII text contains approximately 400 words.

In order to evaluate the automatically generated recommendations we asked additionally for matching recommendations by the human actors. We collected three different types of human judgments on the similarity of interests, skills and expertise among the participants.

First, we asked each of the participants to rank the other 17 members of the network with regard to their similarity in interests, skills, and expertise with himself or herself. When judging, the individual actors did not know the documents released by the other participants. So they had to fully rely on their mutual knowledge. We assumed that these ratings would suffer from the fact that some members of the network were not perfectly familiar with each other.

For this reason, we got two additional rankings from a central node in the network who knew all the other participants rather well. The first ranking was just based on his knowledge about the other participants. The second ranking was conducted after the central node took a detailed look into the documents released by the different participants. The latter ranking appeared to be important since the documents released by the different participant didn’t necessarily reflect the whole spectrum of their interest, skill or expertise.

So there were three human generated and one computer generated ranking that were to be compared. Formally, a ranking (no matter whether it is given by computer or human actor) is given by an ordered list of all the participants (except the rankings creator) in descending order, starting with the best fitting person and ending with the least fitting one. In order to compare that kind of information we chose the City-Block-Metric [4], which is quite a simple method of measuring the ‘distance’ between two rankings. The overall distance between rankings is computed by summing up the results of each single comparison.

In detail, for each participant i the following distance value between two different rankings X and Y is calculated:

$$d_{XYi} = \sum_{r=1}^n |x_{ir} - y_{ir}|,$$

where n is the number of participants and $X_i = (x_{i1}, \dots, x_{in})$ and $Y_i = (y_{i1}, \dots, y_{in})$ contain the values of the two rankings. By definition, $x_{ii} = y_{ii} = 0$ for all values of $i \leq n$, meaning that the participants were not asked to assess themselves which would (in general) be a perfect match. Finally, the overall distance between the two rankings is computed by summing up the distance values for each participant and dividing it by the number of participants to create the average.

To gain a better understanding for the calculated distances, we additionally introduced the distance value for an ‘arbitrary recommendation’. Such a value can be calculated by computing the sum of all possible mismatches (distances) divided by the overall number of possible combinations. The result can be seen as the *expected value* of an overall error when a ranking is created at random.

As an example, let us assume that there are 4 participants: Regarding the assignment for the first rank, an error (distance) of 0, 1 or 2 can occur since each of the participants has only 3 other participants to assign. Here, the error cannot be larger than two in the case 1 being correct and 3 being assigned (or vice versa). In case of the other possible assignments of two and three the error

values are computed in the same way leading to an expected value for the overall error of:

$$0,889 = \left(\frac{0+1+2}{3} + \frac{1+0+1}{3} + \frac{2+1+0}{3} \right) / 3$$

In our case of 18 participants the expected value of the overall error for an arbitrary assignment is 5.647 which can be proved easily. When regarding the results we can assume this value as being a key value which the computer generated assignment should clearly fall below.

4.2 Results

In the following section, we want to present the results of the empirical evaluation. Figure 4 displays the average City-Block distances for each possible combination of assignments. Additionally, the expected distance for an arbitrarily generated ranking is also shown (bottom row) as a constant value since it does not depend on any of the other judgements. We refer to the computer generated assessments as *C*, to the self-assessments as *S* and to the central nodes ('experts') assessments without and with insight to the documents as *E0* and *E1* respectively.

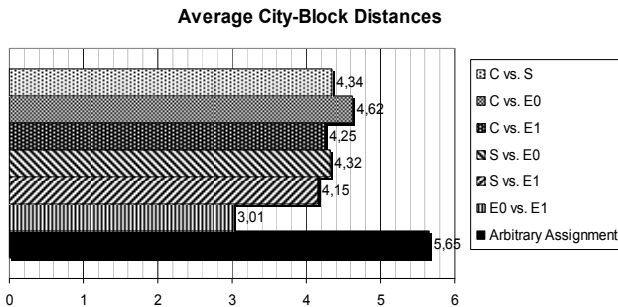


Figure 4: Resulting average distances for each possible combination of assignments

We will first have a look at the first three values that represent the distances between *C* and each of the human generated assignments. Obviously, the ranking *E1* is most similar to *C*. The self-assessment *S* of the participants is second. The central node's first ranking *E0* follows, under the condition that he did not know the actors' released documents.

The data of the participant's self-assessment had some problems. Some of the participants were obviously not sufficiently aware of each others interests, skills, or expertise. Therefore 8 out of the 18 participants provided incomplete rankings, typically missing the last few ranks. Since the City-Block method requires complete rankings, we supplemented the incomplete rankings by assigning the missing ranks at random.

To better understand the values presented in figure 4, we have additionally made a comparison between the three different rankings created by human actors (rows 4-6 in fig. 4). It turned out that the differences among the ratings created by humans (except between *E0* and *E1*) are located in the same range as those in the first three rows. In more detail: The distance between *S* and *E1* has a value of 4.15 which is slightly smaller. The least of the distances measured was the one between *E0* and *E1* which is likely to be a result of both assignments being created by the same actor.

4.3 Discussion

When regarding the results of the evaluation we should keep in mind that we compared four assignments with each other: Three of them are provided by human actors and one is generated automatically. Since it is difficult to exactly measure and quantify human attributes like interests, skills or expertise 'objectively', we did not have any fixed point of reference to compare the different assignments with. However, we feel, that the self-assessments (*S*) given by the users and the one made by the supervisor (expert) after insight of the documents (*E1*) are likely to be good references for the computer generated one (*C*).

The diagram shows that the distances between the computer generated assignments and each of the three human generated assignments in fact fall below the arbitrary assignment – our minimal requirement – but still they remain remarkably high. It is obvious that the computers recommendation is slightly closer to the *E1* than to *E0* which may be explained by the fact that insights into the documents have guided both, the supervisor as well as the system. Given that most of the distance values for comparisons between *S*, *E0* and *E1* are located in the same range as *C*, we can still be satisfied by the results that are presented here. These results also show that measuring 'similarity' between human actors is a highly complex undertaking.

Taking a closer look at the valid single results of the comparisons for each of the participants we observed that most of the distances are fairly close to the average distances between *C* and *S* (4,34). Figure 5 shows these single results for the comparisons of *C* against *S*, *C* against *E0* and *C* against *E1* (the other values were left out in order to avoid complexity). It should give an impression of the distribution of the single distance values as well as the deviations from the average⁴.

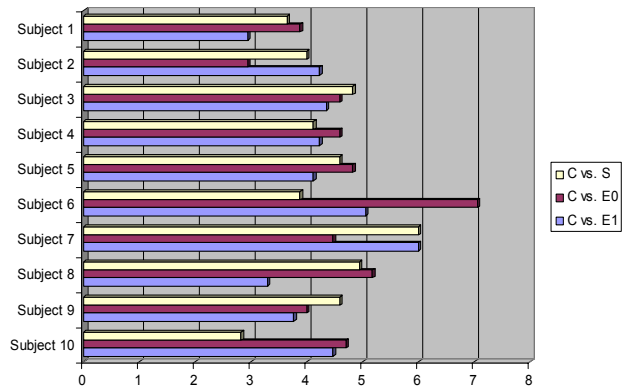


Figure 5: Single results of the distance measurements for C vs. S, C vs. E0 and C vs. E1

Obviously, the subjects 6 and 7 show the highest deviations from the average value. In case of subject 6 it is the value of *E0* only that is clearly above the average. Apparently, the assessment has been improved slightly after having an insight to the documents (*E1*) but is still one of the largest values. In case of subject 7 two of distances are remarkably high. This single result – which was

⁴ The standard deviations for *C* vs. *S*, *C* vs. *E0* and *C* vs. *E1* are 0,858, 1,058 and 0,860 respectively.

far away from our expectations – additionally increases the average distance. To explain this phenomenon, further investigations are needed. An answer could be found in the documents we were given by that person: If those had a weak relation to the author, the system could have been ‘mislead’.

Though the latter case appears to be an exception within our evaluation, we still are aware of the resulting distances being considerable. Some factors are likely to influence the results:

- Both, the self-assessments as well as the central node’s assessments are subjective judgments. Obviously, human judgements on interests, skills or expertise vary considerably, as well. The remarkable high distance of 4,15 between S and E1 indicates this.
- The relatively small set of documents which the participants provided is not likely to represent the entire spectrum of an individual’s interests, skills or expertise. In contrast, human judgements are based on a much more complete impression of the other persons work.
- Sets of documents from different languages are likely to contain disjoint subsets of (key-) terms even though they deal with the same domain. In our case, a large amount of documents was written in English (81) whereas the rest of them were written in German (38). This fact definitely creates problems for the algorithmic matching approach.

Summing up the evaluation shows that the automatic generated assessments derived from the users’ text production are promising. Firstly, the measured distances have visibly fallen below an arbitrary assignment. Secondly the distances that were measured between the different human assessments (except between E0 and E1) were within the same range of variation. The TABUMA approach offers several opportunities for further improvement. Problems caused by texts given in multiple languages may be avoided by multiple user profiles (one for each language). Moreover, we assume that the automatically generated assessments become more valuable in case the mutual knowledge about each others work decreases – which often is the case in larger organizations. This certainly is an issue of further investigations. Furthermore, it is possible to add time stamps to the recurring user profile updates that would also reflect possible dynamics in the actors’ working focus.

5. CONCLUSIONS

As a result of the increasing acceptance of knowledge being an important factor for productivity gains and innovation in organizations, the importance and the value of KM has increased. Since knowledge in terms of experience, abilities or interests is inherently bound to human actors and thus cannot be externalized, expertise recommender systems have surfaced, focussing on connecting human actors. One of the key challenges with regard to these systems is the question of how to identify ‘carriers of knowledge’. Even though we can think about several other useful sources of personal data, we believe that text documents are a good indicator for the domains of knowledge associated with a specific human actor. The results of our investigations support this thesis.

In this paper, we describe our approach of user profiling and match-making based on document analysis. We also present

results of an evaluation study. The system generates user profiles out of text documents by means of linguistic methods for extracting keywords, and for the match making *latent semantic indexing* is applied. The TABUMA approach should be easily applicable in organizational contexts since only standard technology and common file formats are required. The handling of text in different natural languages is supported as well. However, our approach could still be refined by creating ‘multilingual profiles’ (see above).

A first implementation provided the base for an empirical evaluation. The evaluation study has confirmed that the approach is promising. Measuring the quality of the recommendations turned out to be a sticky problem since no ‘objective’ baseline recommendation exists. In such a situation, our specific approach of comparing human- and computer generated recommendations seems to be a suitable and practical way. Using the City-Block-Metric in order to measure the quality of those automatically generated recommendations by comparing against human assessments appeared to be a simple and advisable but still straightforward way. Maybe there are more sophisticated ways of quality measurement. However, the empirical evaluation within the research network indicates that variations between the automatically and human generated recommendations were in the range of variations between recommendations of human actors.

From our experiences we can denote some limitations of our approach: First the existence of a collection of meaningful text files is required which is indicative for human actors’ interests, skills or expertise. This is certainly given in many – but not all – possible fields of application. For instance, in certain shop floor settings human actors’ text production or consumption may not be the best indicator for the workers’ interests, skills or expertise. Another point is the ratio between keywords and collocations – under different circumstances (e.g. the amount of text for creating user profiles) this ratio may vary for better results. Therefore this value is still a matter of further investigations.

Further investigations are also needed in order to identify factors that affect the performance of workgroups that have been created by a recommender system based on text document analysis. Certainly other attributes than the persons skills and abilities only must be taken into account.

Apart from matching different users based on the similarity of their interests, skills or expertise, the algorithmic framework of the TABUMA approach can be applied in an additional way. A search engine can be built which delivers hints to human experts as well as to specific documents. Contrary to state of the art search engines, such as Google, arbitrary text-based inquiries could be efficiently dealt with. A user’s text-based inquiry could be converted into a profile vector [25]. Such an inquiry profile could then be matched against the profiles of human actors or given documents. Combined with the application presented here, an interesting tool set for expertise sharing [3] would be provided.

6. ACKNOWLEDGMENTS

We thank Jochen Battenfeld for implementing large parts of the prototypic system and accomplishing the evaluation [6]. We also thank Radhakrishnan Subramaniam for his contribution for making valuable comments, proof reading, and for creating the final version of this paper.

7. REFERENCES

- [1] Ackerman, M. S. and McDonald, D. W. Answer Garden 2: Merging Organizational Memory with Collaborative Help. In *International Conference on CSCW '96*, ACM Press, New York, 1996, 97-105.
- [2] Ackerman, M.S. Augmenting Organizational Memory: A Field Study of Answer Garden, *ACM Transactions on Information Systems (TOIS)*, vol. 16, no. 3, 1998, 203 – 224.
- [3] Ackerman, M. S.; Pipek, V.; Wulf, V. (eds): *Expertise Sharing: Beyond Knowledge Management*, MIT Press, Cambridge MA 2003.
- [4] Backhaus, K., Erichson, B., Plinke, W., Weiber, R.: *Multivariate Analysemethoden – Eine anwendungsorientierte Einführung*. Springer, Berlin 1996.
- [5] Balabanovic, M and Shoham, Y. Fab: Content-Based, Collaborative Recommendation. In: *Communications of the ACM*, vol. 40, no. 3, 1997, 66-72.
- [6] Battenfeld, J. *Benutzer-Matching auf Basis automatischer Textanalyse. Ein Ansatz zur Ähnlichkeitsbestimmung von Benutzern durch Dokumentenanalyse für das ExpertFinder Framework*. Diploma thesis, University of Siegen, 2005.
- [7] Becks, A.; Reichling, T.; Wulf, V.: Expert Finding: Approaches to Foster Social Capital, in: *Huysman, M.; Wulf, V. (eds.): Social Capital and Information Technology*, Cambridge MA 2004, 333-354.
- [8] Berry, M.W., Dumais, S. T., and O'Brien, G. W. *Using linear algebra for intelligent information retrieval*. SIAM Review, vol. 37, no. 4, 1995, 573-595.
- [9] Cohen, D. and Prusak, L. In Good Company: How Social Capital makes Organizations Work, *Harvard Business School Press*, Boston, 2001.
- [10] Deerwester, S., Dumais, S., Furnas G., Landauer, T. and Harshman, R., Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, vol. 41, no. 6, 1990, 391-407.
- [11] Hinds, P. J. and Pfeffer, J. Why Organizations Don't "Know What They Know. In: Ackermann et al. *Sharing Expertise - Beyond Knowledge Management*, MIT Press, Cambridge, 2003, 3-26.
- [12] Heyer, G., Quasthoff, U. and Wolff, Ch. Möglichkeiten und Verfahren zur automatischen Gewinnung von Fachbegriffen aus Texten. <http://wortschatz.informatik.uni-leipzig.de/asv/publikationen/HeyFachbegriffe100902.pdf>, 2002, seen: 20.11.2004.
- [13] Kautz, H., Selman, B. and Shah, M. ReferralWeb: Combining Social Networks and Collaborative Filtering. In: *Communications of the ACM*, special issue, vol. 40, 1997, 63-65.
- [14] McArthur, R. and Bruza P. Discovery of implicit and explicit connections between people using email utterance. In: *ECSCW 2003 – Proceedings of the Eighth Conference on Computer Supported Cooperative Work*, Helsinki, Finland, 2003, 21-40.
- [15] McDonald, D.W. Recommending Collaboration with Social Networks: A Comparative Evaluation. In: *Proceedings of the 2003 ACM Conference on Human Factors in Computer Systems*, Ft. Lauderdale, 2003, 593-600.
- [16] McDonald, D. W. *Moving from Naturalistic Expertise Location to Expertise Recommendation*. Dissertation thesis, University of California, Irvine, 2000.
- [17] McDonald, D. W. Evaluating Expertise Recommendation. In: *Proceedings of the 2001 International ACM Conference on Supporting Group Work*, ACM Press, New York, 2001, 214-223.
- [18] Pipek, V. and Wulf, V. Pruning the Answer Garden: Knowledge Sharing in Maintenance Engineering. In: *Proceedings of the Eighth European Conference on Computer Supported Cooperative Work (ECSCW 2003)*, Helsinki, Finland, 14-18 September, 2003, Kluwer Academic Publishers, 2003, 1-20.
- [19] Porter, M.F. An algorithm for suffix stripping. <http://www.tartarus.org/~martin/PorterStemmer/def.txt>, 1980, seen: 06.11.2004.
- [20] Resnick, P. et al.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, Chapel Hill, North Carolina, United States 1994, 175-186.
- [21] Resnick, P. and Varian, Hal R. Recommender systems. *Communications of the ACM*, special issue, vol. 40, 1997, 56-58.
- [22] Runte, M. *Personalisierung im Internet – Individualisierte Angebote mit Collaborative Filtering*. Deutscher Universitäts-Verlag, Wiesbaden 2000.
- [23] Rucker, J., Polanco, M.J. Personalized Navigation for the Web. *Communications of the ACM*, special issue, vol. 40, 1997, 73-75.
- [24] Snowball Main Page: *German stemming algorithm*. <http://www.snowball.tartarus.org/german/stemmer.html>, seen: 06.11.2004.
- [25] Streeter, L. A. and Lochbaum, K. E. Who Knows: A System Based on Automatic Representation of Semantic Structure. In: *RIAO '88*, Cambridge, MA, 1980, 380 - 388.
- [26] Sullivan, D. *Document Warehousing and Text Mining*. Wiley, New York 2001.
- [27] Terveen, L., Hill, W., Amento, B., McDonald, D. and Creter, J. PHOAKS: A System for Sharing Recommendations. In: *Communications of the ACM*, special issue, vol. 40, 1997, 59-62.
- [28] Tyler, J. R., Wilkinson, D. M. and Huberman, B. A., Email as spectroscopy: automated discovery of community structure within organizations, in: Huysman, M.; Wenger, E.; Wulf, V. (eds): *Communities and technologies 2003*, Kluwer, B.V, 81-96.
- [29] Vivacque, A. and Lieberman, H. Agents to assist in finding help, In: *Proceedings in the Conference on Human Computer Interaction (CHI 2000)*, ACM Press, New York, 2000, 65-72.
- [30] Wenger, E. *Communities of Practice*, In: *Cambridge University Press, Cambridge*, 1997.