Advanced Personalization and Document Retrieval Techniques in Support of Efficient Knowledge Management

J. Vertommen[1], F. Janssens[2], B. De Moor[2] and J. Duflou[1]

[1]Centre for Industrial Management, Katholieke Universiteit Leuven, Leuven, Belgium

[2]ESAT-SCD, Katholieke Universiteit Leuven, Leuven, Belgium

Abstract

A problem that is often encountered by any innovation driven company is the so-called reinvention of the wheel. Effort and money are invested in the development of solutions and technology that are already available within the same company. On average 19% of the conceptual phase in the development cycle is spent on information lookup. A better knowledge both of previously performed research as well as of the availability of expert employees can help solve this problem and would allow shortcuts in the development cycle.

This paper describes some techniques from the field of knowledge discovery and data mining that enable efficient searching for existing expertise and knowledge. These techniques are based on documents found on a company's intranet and require no extra effort from the end user, such as the construction of taxonomies. The scope of the functionalities derived from these techniques however far exceeds document retrieval facilitation.

One technique is the creation of user profiles. Such a profile captures the expertise of an employee. Another technique is document searching based on LSI (Latent Semantic Indexing), offering significantly better results than traditional search algorithms. Finally, both user profiling and LSI can be applied in combination with each other, using a user profile in latent semantic space as background for the document or user search algorithm.

The described techniques were tested in a realistic R&D environment and conclusions are based on these empirical results.

Keywords: Knowledge management, user profiling, LSI

1 INTRODUCTION

During the last few decades, the industrial world has witnessed a growing awareness of the importance of knowledge. Knowledge has been recognized as a dominant economic factor to any company. This is not at all surprising. The value of a company is only partly represented by it's tangible assets such as buildings, equipment and capital. A product is only elevated to a good product by the knowledge and experience of the people that design and produce it. This knowledge is often tacit, i.e. ungraspable, residing in people's heads, but it can also be made explicit, for instance by writing it down in product manuals [1].

The ultimate goal is to manage this knowledge in the most effective way. Many technologies exist to achieve this, here generally called knowledge management systems (KMS). Two typical problems that arise when knowledge is not managed (efficiently) are the inaccessibility of said knowledge to others and, as a result, the phenomenon of reinventing the wheel. Time studies indicate that throughout a design process designers spend on average 19% of their time - with peaks to 36% in early conceptual design phases - on information gathering activities [2][3]. An efficient knowledge management system that allows advanced searching in the existing knowledge base of a company could seriously reduce these percentages. This unavailability of knowledge takes on dramatic proportions when product designers have to start from scratch, ignorant of research that has been performed earlier by people in the same company. Very often, especially in large companies, employees have no idea what other project groups or divisions are working on, let alone the exact details of the techniques or products they design. A survey by KPMG talks about 63% of companies complaining about this reinventing of the wheel, as opposed to 45% of those that have implemented some kind of KMS [4].

Managing tacit knowledge can be done by bringing people with similar interests and experience together using a recent concept called Communities of Practice [5]. These CoP's surpass company or project boundaries.

Techniques for managing explicit knowledge have been known for much longer, drawing from the fields of data mining, knowledge discovery and artificial intelligence. As stated earlier, explicit knowledge is tangible in the form of product descriptions, manuals, journal papers, figures, meeting minutes and so on. These data can be analyzed for certain characteristics and correlating features using statistical or artificial intelligence techniques. These techniques on their part enable functionalities for KMS's. This paper will discuss two such techniques: user profiling and Latent Semantic Indexing, and the functionalities that are enabled by incorporating them in a KMS.

2 USER PROFILING

User profiling is commonly used to model the interests and preferences of a user while he is using an application that allows for differentiation between users. Probably the most simple example of user profiling is a cookie containing a

choice of language while surfing to a multilingual website. In a more advanced system, user profiling might be done by giving the user a series of options to choose from to describe himself. Another possibility is to explicitly ask the user to build a profile of himself by entering keywords into the system that he believes to be characteristic. These profiles can then be used as background when performing, for instance, a search with a search engine. The query that was entered into the engine is augmented by the profile of the user, resulting in more accurate search results.

The drawback of the above mentioned methods is that they are

- inaccurate: describing yourself with keywords is a very difficult task and often not adept to information retrieval situations.
- limited: a fixed list of descriptive terms is usually not enough to describe everyone with enough detail

One reason explaining the popularity of these limited technologies is the rise of the Internet. User profiling has been studied for quite some time ([6] is considered as one of its first appearances), but with the emergence of the Internet, focus shifted to this medium. Characteristic of surfing the net however, is the volatile way in which it is performed. A user is mostly only known for the duration of one session on a certain website. During this session information on the user can be gathered, but it is lost once the user leaves the website.

In a KMS that runs on a corporate intranet however, the users are fairly fixed, being the employees. Their profiles can be built and tuned during multiple sessions. And as such, their profiles can grow and change over time. As mentioned in the introduction, these KMS's can run on knowledge that has been made explicit in the form of documents. In such a situation, it is recommended to construct user profiles based on documents in the system.

2.1 User profiles based on documents

The idea behind profiling based on documents is that each user of the KMS is characterized by what he reads or writes. When a document is known to be read by a user, this document contains interesting information to this user. When a document is written by a user, and entered into the KMS, it is not only of interest to this particular user, but also illustrates some degree of expertise in the domain covered by the document.

Basically, the process of building user profiles based on documents works as follows:

- each document in the system is analyzed for the words it contains and how many times each word occurs in the document
- based on the total vocabulary of all words contained in the entire document collection, as well as knowledge on the words in each document gathered in the previous step, each word in each document is assigned a certain weight in that document using a TF-IDF weighting scheme [7]
- once this weighting has been achieved, each document can be represented as a vector in the multidimensional vector space, where each dimension is a word from the vocabulary. A document vector has a weight in each dimension; either the weight assigned to it by the TF-IDF weighting scheme in the previous step, or 0 because the word does not occur in this particular document
- since the documents are now represented as vectors, it becomes possible to manipulate them, for instance adding them together. This is what happens when building a user profile; the vectors of documents that are of interest to the user are combined

The result of this process is that each document in the system is represented as a vector, and each user profile is represented as a vector in the very same vector space. This makes it possible to compare the vectors. A distance measure can be defined between vectors (e.g. the cosine distance) enabling a series of functionalities.

It is important to point out that there is room for optimization with these vector profiles. Suppose an employee who has been with the company for several years has to be profiled. The chance of this person having just one expertise is unlikely. It would be a big improvement to assign this person multiple profiles, one for each interest or expertise. Each of these profiles will then be based on documents related to the expertise in question. Automatically identifying such multi-vector profiles is not an obvious task and is beyond the scope of this paper.

2.2 Functionalities

User profiles offer several possible functionalities, some of which will be discussed here. The basis of these functionalities is the fact that employee-profiles as well as documents reside in the same vector space.

Advanced searching

Advanced searching is the first application that springs to mind when thinking of user profiles. Imagine a search engine such as Google™ with access to tons of information, possibly the intranet of some company. Suppose the user of such a system is interested in information on robots. A search on the term 'robot' would result in a huge set of results on all types of robots. But suppose this user has a medical background and is only interested in the application of robots in surgery. If a user profile is available, representing the medical background of this person, the original query 'robot' can be augmented with the user profile.

The query can be expanded as follows:

$$q' = \alpha p + (1-\alpha)q \qquad (1)$$

with q the original query 'robot', p the vector profile and α the weight with which the profile is added to the original query. The result is a new query, q', containing information from the user profile.

When this query is used as input for the search engine, it should return information on robots used in medical applications.

User clustering

Another functionality offered by user profiles is that it is now possible to cluster users together. Based on the vectors representing company employees in the vector space and the similarity between these vectors, groups of employees can be distinguished. The grouping together of several employees suggests that they have an interest in common as can be seen in the following experiment [8].

A set of documents was collected from each of 10 employees at an R&D company. The document sets varied from 8 to 34 in size. From these documents a user profile was constructed for each of the employees. Besides documents, function descriptions of the test subjects were collected, from which a part of the company structure could

be reconstructed. More precisely, a distinction was made between the R&D department and the IT department.

The user profiles were fed to a hierarchical clustering algorithm that exploits the distance between the profile vectors. This algorithm gave as output the dendrogram seen in Figure 1.

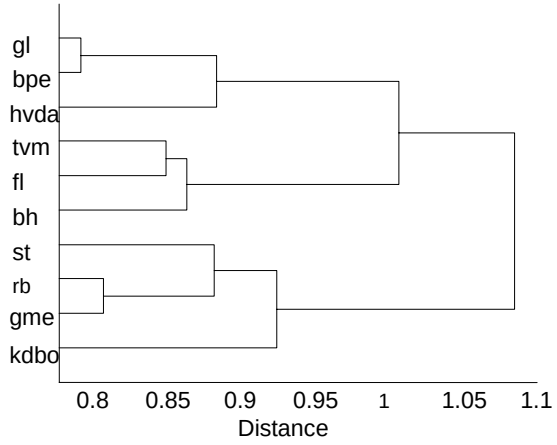

Figure 1: Dendrogram showing the cluster-evolution starting with 10 individual profiles and ending in 1 cluster containing all profiles. The abbreviations in front of each branch are the names of the employees in the dataset. The abcis value is a measure for the distance between clusters.

This dendrogram shows the evolution of the clusters being formed, starting with 10 clusters, each containing one profile (at 0.79 or less on the horizontal axis). One by one the profiles are grouped together in clusters, resulting in one big cluster at x-value 1.09. It is clear how, halfway the clustering process, three distinct clusters are formed (shown in Table 1).

Cluster	Members
A	gl, bpe, hvda
B	tvm, fl, bh
C	st, rb, gme, kdbo

Table 1: The 3 main clusters

When comparing the formed clusters with the company structure, the algorithm appears to have grouped together people who can be expected to have the same interests and/or competences;

Cluster A contains employees that work in the Research and Technology Development division. Cluster C on the other hand contains employees of the IT division. Cluster B required some more thorough analysis since the clustered employees are affiliated with different functional units of the company. When examining the data more in detail it became clear that these were higher-ranked employees and that they were clustered together on the fact that they have more general management interests than specialized technological skills.

This structure is reflected when investigating the most important words (here reduced to word-stems) on which profiles are grouped together (See Table 2).

Cluster	Stems
A	fast, damp, modal, figur, structural, experimenta, sine, vibraat, …
B	effici, compani, relationship, organisaa, longer, sale, memo,…
C	dataserv, intern, studi, extern, emb, client, xml, securiti, web, intranet

Table 2: Dominant word stems per cluster

2.3 Conclusions

The integration of user profiles in a KMS makes it possible to take into account the expertise of employees in a company. This offers significant improvements in applications that are used to search for information in the form of

- faster results: the user profile helps to guide the search engine towards interesting information
- better results: the returned results will be in relation to the user profile, i.e. in the context of the user's expertise

A second functionality offered by user profiles is the ability to group users together, discovering relationships that might have gone by unnoticed otherwise. These user groups can be used for:

- building the CoP's mentioned in the introduction
- guiding employees in search of information to the correct expert
- pushing information towards employees based on their relationship with others and the interests of these others

3 LATENT SEMANTIC INDEXING

Up to this point a single document or a single user profile has been represented as a vector with as many dimensions as the number of distinct terms (stems) in the vocabulary of the complete document set. The weight of each term in the vector is calculated according the TF-IDF weighting scheme [7]. When one considers the whole document collection, all document vectors can be combined to form a 'term by document matrix' $\boldsymbol{A}$ that contains in each column the vector of one document. The number of columns in the matrix $\boldsymbol{A}$ is thus equal to the number of documents in the set and the number of rows equals the number of terms in the vocabulary. In large document collections, the global vocabulary easily spans tens of thousands of words. Since such a high dimensionality severely compromises computational performance, a reduction of the dimensionality is desirable.

3.1 Theory

Latent Semantic Indexing is a mathematical technique based on the truncated singular value decomposition [9] of a matrix, that analyzes the term by document matrix $\boldsymbol{A}$ to find the major associative patterns of word usage in the document collection and to get rid of the 'noise variability' in it. It assumes that there is some underlying or latent structure in word usage that is partially obscured by variability in word choice [10][11]. LSI makes it possible to compose another matrix $\boldsymbol{A_k}$ that is the best approximation of $\boldsymbol{A}$, but with rank k much lower than the column or row dimension of $\boldsymbol{A}$. Typical values for k range from 100 [10] to about 300 [12]. Instead of the huge number of rows in the

matrix $\boldsymbol{A}$ (equal to the total number of terms), only k statistically derived orthogonal indexing 'factors' or 'concepts' remain in $\boldsymbol{A_k}$. The vocabulary of the document set is thus replaced by a much smaller set of concepts that are talked about in the document set. A document has a weight for each concept that indicates how much it is represented in the document. If the weight is highly positive, it means that the document has much to do with that concept. A negative weight on the other hand indicates a negative correlation, meaning that the document is not about the corresponding concept.

3.2 Functionalities

The interesting effect of Latent Semantic Indexing is the fact that synonyms or different term combinations describing the same concept will be mapped onto the same factor based on the common context in which they generally appear, even for terms that do not co-occur in any document. Besides the implicit relating of synonyms, also the problem of polysemy (a word with different meanings) is addressed by LSI.

In an information retrieval scenario where a search engine is used to find documents interesting for a certain query, the main advantage of LSI is that documents will be found even if they do not literally contain the query words. E.g. the query 'car' might also retrieve documents that only talk about 'automobile'. And this without the use of any form of dictionary.

4 USER PROFILES IN LATENT SEMANTIC SPACE

A user profile vector can also be represented in latent semantic space. Such a latent semantic profile is a conceptual representation of the interests or competences of a person, rather than a weighted collection of literal terms. Mathematically, a user profile vector can either be projected in the latent semantic space by techniques from linear algebra (the same way as a search query is projected in LSI space before comparing it to every document), or it can be built up by just combining the latent semantic vectors of the documents on which it is based. The same techniques for combining user profiles and document vectors (formula (1)), calculating the similarity between documents or user profiles, and the clustering of documents and users, can be applied on the latent semantic vectors.

A KMS will benefit from the same advantages of LSI as a search engine does, e.g. a more efficient searching of the knowledge base and experts. Tests have shown that the performance (precision) of a search engine that uses LSI is about 30% better than one that does not [10]. This increased retrieval performance helps to attain a better knowledge both of previous research and the availability of expert employees and can thus shorten the development cycle.

5 CONCLUSIONS

All techniques described in this paper were validated in an R&D environment, both academic and industrial.

As concluded in 2.3, the constructed user profiles proved to be representative for the user's expertise, enabling a number of functionalities described in 2.2.

As results of the techniques involving LSI mostly concern the improvement of personal user experience, they are harder to quantify. Preliminary experiments using a latent semantic space on gathered R&D data show a promising effect though. These expectations are confirmed by the results of others, mentioned in 4.

6 ACKNOWLEDGMENTS

Research supported by Research Council KUL: GOA-Mefisto-666; FWO: G.0407.02; IWT: GBOU-McKnow; Belgian Federal Science Policy Office: IUAP V-22 .

7 REFERENCES

[1] Nonaka I., Takeuchi H., 1995, The Knowledge Creating Company, Oxford Press, ISBN 0195092694

[2] Ullman D., Dietterich T., Stauffer L., 1988, A Model of the Mechanical Design Process Based on Empirical Data, Artificial Intelligence in Engineering Design and Manufacturing, 2/1, pp. 33-52

[3] Court A., Culley S., McMahon, C., 1993, The Information Requirements of Engineering Designers, Proceedings of ICED 1993, The Hague, August 17-19, pp. 1708-1716

[4] KPMG, 2000, Knowledge Management Research Report

[5] Wenger E., Communities of Practice: Learning, Meaning and Identity, 1999, Cambridge University Press, ISBN 0521663636

[6] Perrault C.R., Allen J.F, Cohen P.R., 1979, Speech acts as a basis for understanding dialogue coherence, Report 78-5, Department of Computer Science, University of Toronto, Canada

[7] Salton G., McGill M.J., 1986, Introduction to Modern Information Retrieval., McGraw-Hill, Inc., New York, NY.

[8] Vertommen J., Vandermeulen B., Duflou J., Van Dromme D., De Moor B., 2004, Testing Methods and Algorithms for the Next Generation of Knowledge Management Systems, Proceedings of the FUBUTEC 2004 Conference on Future Business Technology, Fontainebleau, France, pp. 15-19.

[9] Golub G. H., Van Loan C. F., 1983, Matrix Computations, Johns Hopkins University Press, Baltimore, Maryland, 1983.

[10] Deerwester S., Dumais S. T., Landauer T. K., Furnas G. W., Harshman R. A., 1990, Indexing by latent semantic analysis, Journal of the Society for Information Science, 41(6), pp. 391-407.

[11] Berry M. W., Dumais S. T., O'Brien G. W., 1995, Using linear algebra for intelligent information retrieval, SIAM Review, 37(4), pp. 573-595.

[12] Berry M. W., Dumais S. T., Shippy A. T., 1995, A Case Study of Latent Semantic Indexing, Technical Report, University of Tennessee.