Integrating different Machine Learning methods to support search in cross-domain information sources - the project AWAKE

Michael Wurst*, **Jasminko Novak****and **Martin Schneider*****

*University of Dortmund, Artificial Intelligence Department, D-44221 Dortmund email: wurst@ls8.cs.uni-dortmund.de

** Fraunhofer Institute for Media Communication, MARS Exploratory Media Lab, Schloss Birlinghoven, D-53754 Sankt Augustin, email: Jasminko.Novak@imk.fhg.de

*** Fraunhofer Institute for Media Communication, MARS Exploratory Media Lab, Schloss Birlinghoven, D-53754 Sankt Augustin, email: Martin.Schneider@imk.fhg.de

Abstract. In this paper we will present some of the ongoing research in the project "AWAKE - Networked Awareness for Knowledge Discovery". The aim of this project is the development of models for interactive information search in heterogenous data sources and the exchange of knowledge within expert communities. For both purposes, Machine Learning takes a major role. Our approach is to integrate supervised and unsupervised learning on the one hand, and content and interaction analysis on the other hand. By integrating multiple approaches into a single system, users can choose freely between different search strategies. Through the combination of content and interaction analysis users can benefit not only from the data provided by the system, but also from the work and knowledge of the other users. We will show, how all these approaches can be centered around one common concept: the knowledge map.

Keywords. information search, Machine Learning, collaborative methods, Knowledge Management

1 Introduction

In the past, there have been several attempts to employ Machine Learning methods to deal with the problem of information overload in large information spaces.

One basic approach is to use Machine Learning to impose a structure on a previously unstructured information space and to give the user the possibility to navigate it. This enables the user to apply more efficient search strategies than just to go through all items sequentially. A typical example for the use of unsupervised learning to gain such a structure is the "scatter/gather"[3] approach. A clustering algorithm is used to group documents together, which contain similar terms. This procedure is applied recursively at different levels, enabling users to employ tree search like strategies by picking at each level the clusters, which seem to contain the desired information most likely.

Another approach to use Machine Learning to support information search is based on the assumption, that users have specific information needs or preferences, which they are either not able to express explicitly or which would be costly to express. In such cases a system can try to discover these information needs automatically and then provide the users with corresponding items. In the "relevance feedback"[12] approach, the system iteratively refines a search term given by the user according to the relevance judgments on documents in the result set of each cycle. Another example is the system described in [2], which has the purpose of providing the user with interesting web pages. The user judges the "interestingness" of the pages presented to him and the system then uses these judgments to incrementally learn an interest profile of the user.

All the systems mentioned so far do not make use of the fact that usually many users interact with an information system. Though past searches can include valuable information for future searches, as people often have similar interests and information needs. Consequently, Machine Learning methods can not only be applied to the information space itself, but also to past interactions in this information space. Perhaps the most widespread class of systems applying this principle are "collaborative filtering" systems (e.g. [11]). Basically users rate items, such as movies. The assumption is, that users which rate a set of items similarly, have similar interests. Accordingly, the systems recommends items to a given user, which users with similar interests have given a high rating. Another approach is realized in the system "WebWatcher"[8], which uses feedback on the successfulness of the past searches to learn to guide users through a hyperlink structure.

Our aim of research within the AWAKE project is to integrate several methods in a single system, which allow the user to explore an automatically structured information space interactively, but which are also capable of capturing the users specific interests or information needs and that offer ways, by which a given user can profit from the work and knowledge of other users. Such an integrated approach offers several advantages. Firstly, the user can always choose the strategy, which seems best suited to him in a given situation. Secondly, from the point of view of Machine Learning, combining content and interaction analysis can help avoiding problems, which arise from using only one of these techniques in isolation. This point will be discussed in more detail later. Thirdly, having different approaches integrated in a single system, makes the comparison of their usefulness much easier. In this sense AWAKE can be seen as an experimental framework to explore and to compare all different kinds of approaches to support search and exploration of complex information spaces. Beside the Machine Learning methods, these approaches also include innovative visualization and interface techniques, which will be mentioned only briefly here. For a broader view of the project refer to [10].

The remainder of this paper is structured as follows. Section 2 gives an overview on the aim and the structure of the system and introduces the concept of "knowledge maps" as basic medium of interaction. In section 3 the individual machine learning methods employed are described in more detail. As this paper deals with ongoing work, section 4 discusses some plans for future research and development.

2 The basic structure and functionality of the system

2.1 The aim of the system

The aim of the system developed is to provide experts in a given domain with a tool, which allows them to search different domain specific information sources for relevant objects. These objects may include scientific publications, project descriptions, links to web resources, etc. Typically, the system will be used to search for relevant informations on a specific subject, to gain an overview on a whole research area, to find out about recent developments in a given area or to find intersections between different research areas.

As a concrete test domain for our system we use netzspannung.org[4], an initiative to build up a cooperative knowledge source with brings together information from the area of media art and technology. The domain of media art is especially interesting as it often combines research from many different disciplines such as HCI, CSCW, AI etc. This heterogeneity is a challenge, as in these works, even if they deal with similar problems, often quite different terminology is used, which makes it hard for users to see any crossrelations directly.

2.2 Preprocessing

The information provided to the users can generally be stored in different heterogeneous databases. This heterogeneity concerns the representation of the objects, as well as the kind of meta information available about a given object. An important step in building an information system, which works with heterogeneous data sources, is to preprocess and transform the data to get an unified view on the information in these sources. For the mentioned domain of media art, we can build on work from a previous project[4], providing a tool to map the information contained in databases to a common domain specific scheme.

2.3 Knowledge Maps

In order to integrate different methods in an easily understandable and usable way, a common interaction model is needed, around which all these methods are centered. We use the paradigm of knowledge maps for this purpose. We refer to a knowledge map as two-dimensional space, containing individual objects as points. The location of the documents relatively to another can express the relation of these objects. Usually objects located nearby are similar or relevant to each other. Additionally objects can be explicitly grouped together. The user gets detailed information about an object, by clicking on it. There is also the possibility to zoom in order to focus on a special area. Knowledge maps can be automatically created by the system or manually be edited by users. In the latter case we refer to personal knowledge maps. The system provides the following functions to work with system generated and personal knowledge maps:

- *Automatically build knowledge maps from the objects contained in a specific information source*
  The system can automatically generate a knowledge map, containing the objects of a specified information source. The objects are located on

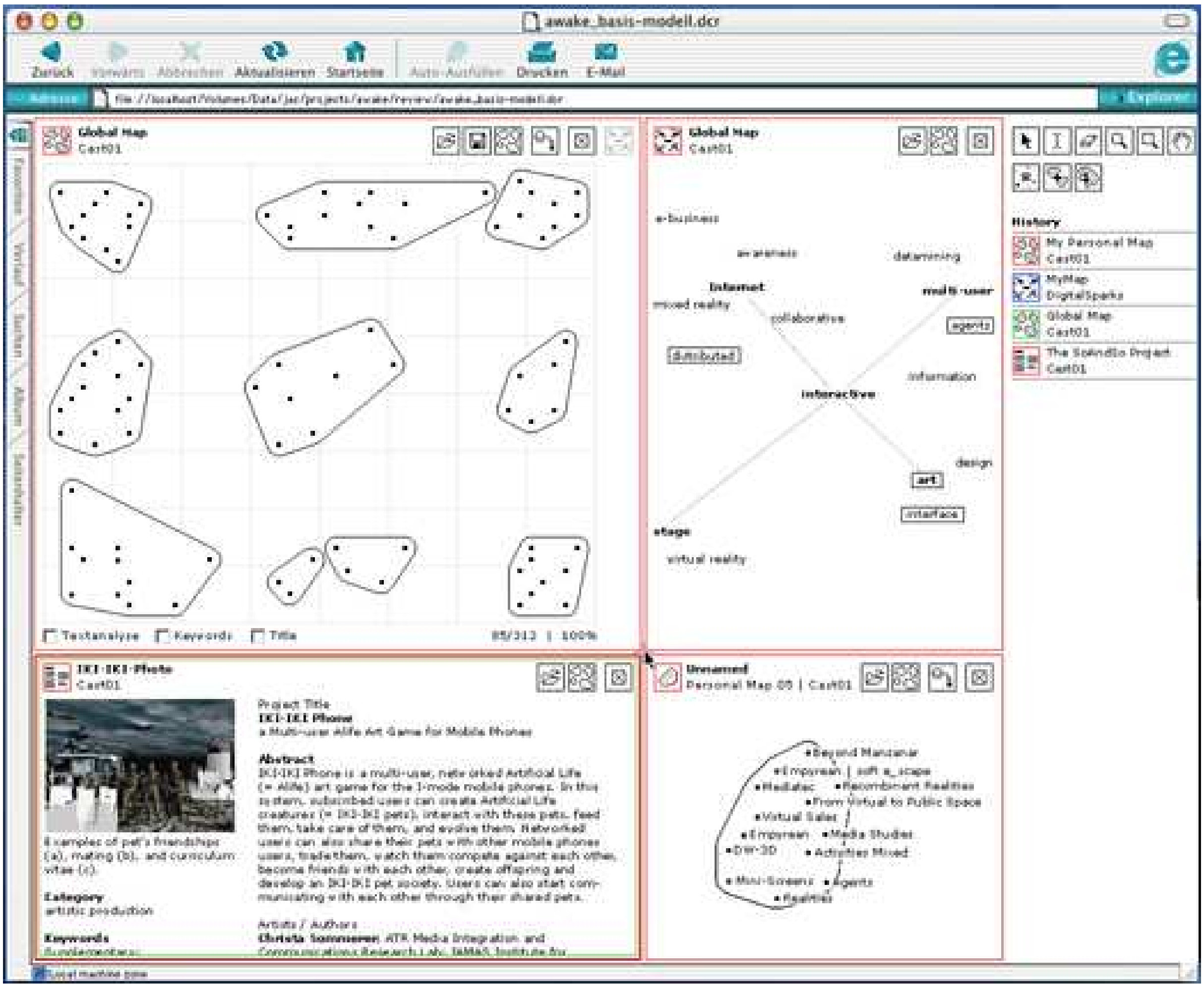

Figure 1 Combined wrapper-based feature selection and generation approach

the map in a way, that similar objects are near by another (Fig 1, upper left). To further support the navigation on this map, automatically extracted keywords are used to characterize different regions of the map. Additionally a second map is created, containing the semantic axis of the objects on the main map. This map, to which we refer as "dimension map" (Fig 1, upper right), can be browsed as well and is used to locate objects, which contain given keywords or to rearrange the main map using only a set of selected terms.

- *Creating personal maps*
  Users can take system generated maps as a starting point to create personal knowledge maps. They do this by dragging objects or whole regions from a system generated map and locate them anywhere on their own map. Objects located on this personal map can be group together by the user and these groups can be labeled. Every user can edit and store several maps.

- *Apply personal maps to a specific information source*
  Beside their use as bookmarks, personal knowledge maps can be applied to a specific information source.The resulting system generated map contains the same groups as the personal map and an additional trash group. Every object in the information source is either assigned to the group in which it fits in best, or to the trash group, if it doesn't fit in any of the groups. By this, the system not only captures the user's information need and provides him with corresponding objects, but also structures these objects in personalized way.

- *Search the set of personal maps created by other users*
  Building a knowledge map, a user imposes a structure on a set of objects. This structure usually contains implicit knowledge of this user and can be the result of a great deal of work. By explicitly publishing such a personal map, he/she

makes this work and knowledge accessible to others. This is a very powerful feature, integrating directly collaborative aspects into the system. To fully benefit from this feature, a simple search function is provided, which allows to find relevant maps efficiently.

Additionally to the functions described here, the systems contains several other features, as an ontology editor (please refer to [10]).

In the following section, the Machine Learning methods used to provide the functionality of automatically structuring a set of objects and of applying a map to an information source, are described in more detail.

3 Machine Learning methods used for generating knowledge maps

3.1 Unsupervised generation of knowledge maps

In order to generate a knowledge map of a set of objects, these objects have to be arranged on a two dimensional area. We use Kohonen's self-organizing neural network[9] for this purpose. As a basis for the application of this technique, we generate word vectors from the textual description of the underlying objects using tf/idf weighting. As the vectors encode semantic properties of texts the map will position semantically correlated texts close to each other.[1]

Additional to the main content map, we employ an approach similar to that described e.g. in [5] to build a "dimension map". The idea is to structure words by examining which other words appear in the context of a given word. The high dimensional context relations resulting from this are then mapped to a two dimensional space, again using the self-organizing network. In this way we can create an initial set of concepts (words) that serve both as an explanation of the clustering and as a navigation structure. Our system provides the additional feature, that users can customize the aspects according to which the maps are generated by manually selecting a number of words on the dimension map. The weights for these words in the vector space are increased making them the "most important" words. Then the mapping procedure is re-applied using these modified weights.

To directly support the navigation on the main map, keywords are automatically extracted using the weights of the self-organizing network. As these weights indicate, which terms are most important for a grid point on the map, they can simplify the orientation significantly.

[1]The use of word vectors as attributes is motivated by the fact, that every object contains at least some kind of textual description, while the remaining object attributes can significally differ, as the objects come from different sources.

3.2 Combining content and context information to learn and apply personal knowledge maps

By creating a personal map, the user defines a set of classes. The idea of learning a personal knowledge map is to find a function, which assign new objects to these classes automatically. After such a decision function has been found, a map can be applied to any single object or information source provided by the system. As it is quite possible, that a given object does not fit in any of the user defined classes, the system provides a special "trash class". The question of whether an object can be reasonably assigned to any of the user defined classes or not is to a significant extent subject to individual preference. As a consequence, the system gives the user the possibility to interactively adjust the threshold of minimal similarity. If there is no object on the personal knowledge map to which the given document is at least as similar as defined by this threshold, the object is assigned to the trash class. Otherwise the decision function is used to assign it to any of the user defined classes. This allows the user to fine tune the personalized classification by exploring the influence of the threshold between two extremes: if the threshold is maximal then all objects are assigned to the trash class, if it is minimal all documents are assigned to some class and trash class is empty.

We have experimented with two different classification algorithms to implement this functionality: Support Vector Machines[6] and Nearest Neighbour[1]. Although SVM perform quite well, for a couple of reasons, we have chosen Nearest Neighbour for our ongoing work. Firstly the selection of parameters is not trivial with the SVM, as the optimal parameters for each personal map can differ significantly[2]. Additionally, we have to deal with a multi class problem in general, which is easier to handle using Nearest Neighbour. Lastly, as will be described in the remainder of this section, we want to use context information for classification, which is most intuitively represented by a similarity measure. This directly suggest the use of an instance based method.

The similarity measure employed for Nearest Neighbour is based on two kinds of information. Firstly on the content of the objects, which is again represented by word vectors as described in the proceeding sec-

[2]Though it would be possible to find optimal parameters automatically by cross evaluation, this would increase the respond time of the system making interactive work quite unpleasant. This is true even if the efficient performance estimators for the SVM presented in [7] are used.

tion. Secondly it is based on the context in which objects appear. The idea is the following: If two objects appear together in many user edited clusters, then we can assume, that these objects are in some way similar. This is a very interesting feature of our system, as items are not only rated by users, like in "collaborative filtering" systems, but are put into the context of other items. This is much more powerful, as usually an item is not interesting or relevant *per se*, but only relevant in a given context.

Both the text-based similarity and the context similarity are in a first step calculated independently of each other. For text-based similarity we use the well known cosine measure:

$$sim_T(x, y) = \frac{\vec{x}\,\vec{y}}{|\vec{x}||\vec{y}|}$$

For context similarity we use the "Dice"- coefficient:

$$sim_C(x, y) = 2\frac{|X \cap Y|}{|X| + |Y|}$$

were X is the set of clusters, which contain object x and Y is the set of clusters, which contain object y.

Using this measure, clusters, which do not contain any of both objects, are not counted, which seems appropriate for the given case. Also co-occurances get double weight, as we consider them as more important than single occurrences. The membership of clusters and objects to personal maps is not taken into account at all, as it is quite unclear, whether objects on the same map, but in different clusters are similar.

The question is, how text based and context similarity should be combined into a single measure, preserving the advantages of both. The advantage of text based similarity is, that it is always applicable and does not rely on user generated data. Though text based similarity can lead to poor results, if the underlying texts are too short or too heterogenous, e.g make use of different terminology or are even written in different languages. On the other, using context similarity, we avoid these problems completely. The disadvantage of context similarity is however, that if only few users add a given object to their maps or if the contexts, in which it appears, diverge, we do not get any reliable evidence on the similarity of this object to other objects.

Consequently, we use a statistical test (chi-square based) to examine, whether the co-occurrences of two objects are significant in a statistical sense. If so, only context similarity is used, as we have a very direct clue of the similarity of these objects. If not, we use only text based similarity, as it works independent of any object occurrences. First experiments on synthetic data show that the combination of both methods is on average superior to any of the methods in isolation. Though only experiments on "real" personal knowledge maps will allow a fully qualified evaluation of the approach.

4 Summery and ongoing work

We have presented an approach of how to use the paradigm of knowledge maps as central concept to integrate different methods for interactive information search. We have shown, how supervised and unsupervised learning can be used to generate knowledge maps, providing users with different views on the content of an information source. Furthermore, we have presented possibilities to use knowledge maps as medium for explicit and implicit exchange of knowledge between different users. As pointed out, our system differs significantly from so called "collaborative filtering" systems, as items are not just rated by the users, but are put into context. In this sense, our system enables "collaborative structuring" rather than just "collaborative filtering".

Though our first results are promising, there is still a number of open issues.

At the moment, we use only two similarity measures for classification: text-based similarity and context similarity. We are planning to incorporate additional ones, based on e.g. authorship, year of publication etc. The combination of such heterogeneous aspects is a non-trivial task, especially because many of the values for these attributes are missing, as they are not available from a given information source or not even definable in a meaningful way. Therefore further research in this direction will be necessary. This also includes the search for new interface techniques, allowing the user to explicitly state, which kinds of similarity are of importance to him/her, as in many situations, this can not be estimated purely on the basis of data.

Another problem is, that we use the context similarity only for supervised learning at the moment. We are investigating, how this kind of information can be incorporated into the methods for unsupervised knowledge map generation as well.

Editing personal knowledge maps, the user can arrange objects only in flat structures, which is very intuitive and easy to handle, but not always sufficient. Therefore the system contains a second editor, capable of creating hierarchical structures and other relations between objects (see [10]). From the point of view of Machine Learning, the problem is to develop such methods, which fully exploit the information contained in such structures.

Finally, an evaluation workshop is planned, to analyze the usefulness of the system and to compare the individual contributions of the different approaches.

5 Acknowledgments

The work described in this paper has been undertaken within the projects "AWAKE - Networked Awareness for Knowledge Discovery" and netzspannung.org - an Internet Media Lab, both financed by the German Federal Ministry for Education and Research. The netzspannung.org project aims at establishing a knowledge portal that provides insight in the intersections between digital art, culture and information technology. It is led by Monika Fleischmann and Wolfgang Strauss at Fraunhofer Institute for Media Communication.

References

1. D. Aha, D. Kibler, and M. Albe. Instance based learning algorithms. *Machine Learning*, 6:37–66, 1991.
2. M. Balabanovic, Y. Shoham, and Y. Yun. An adaptive agent for automated web browsing. *Journal of Visual Communication and Image Representation*, 6(4), 1995.
3. D. R. Cutting, J. O. Pedersen, D. Karger, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329, 1992.
4. M. Fleischmann, W. Strauss, J. Novak, S. Paal, B. Mueller, G. Blome, P. Peranovic, C. Seibert, and M. Schneider. netzspannung.org - an internet media lab for knowledge discovery in mixed realities. In *Proceedings of the Conference cast01 - living in mixed realities*, 1991.
5. T. Honkela. *Self-Organizing Maps in Natural Language Processing*. PhD thesis, Helsinki, Finland, 1997.
6. T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.
7. T. Joachims. Estimating the generalization performance of a SVM efficiently. In P. Langley, editor, *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 431–438, Stanford, US, 2000. Morgan Kaufmann Publishers, San Francisco, US.
8. T. Joachims, D. Freitag, and T. M. Mitchell. Web watcher: A tour guide for the world wide web. In *IJCAI (1)*, pages 770–777, 1997.
9. T. Kohonen, S. Kaski, K. Lagus, J. Salojrvi, V. Paatero, and A. Saarela. Organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11:574–585, 2000.
10. J. Novak, M. Fleischmann, W. Strauss, M. Schneider, M. Wurst, K. Morik, J. Ziegler, and C. Kunz. Augmenting the knowledge bandwidth and connecting heterogeneous communities through uncovering tacit knowledge. In *Proceedings of the IEEE Workshop on Knowledge Media Networking (KMN 2002), Kyoto*, 2002.
11. P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.
12. G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.