

User Adaptive Methods for Interactive Analysis of Document Databases

Andreas Nürnberger¹ and Marcin Detyniecki²

¹ University of California at Berkeley, EECS, Computer Science Division, USA
email: anuernb@eecs.berkeley.edu

² Laboratoire d'Informatique de Paris 6, CNRS, Paris, France
email: marcin.detyniecki@lip6.fr

ABSTRACT: The currently available document retrieval tools are usually still based on simple keyword search methods and according result list or a static (hierarchical) classification of the considered document collection. Both approaches neither allow a user to adapt the ranking or the classification to his needs, nor provide visualization methods to support the user in browsing or analysing a document collection. In this article we give a brief overview of our work on user adaptive methods for searching and browsing in document collections.

KEYWORDS: information retrieval, document classification, visualization, retrieval support system, user feedback

INTRODUCTION

Searching in and especially the analysis of semi-structured or unstructured document databases containing text, image or other multimedia documents is still a problem that is only insufficiently supported by currently available retrieval tools. The existing tools usually offer only simple keyword based search interfaces and provide in the best case either a fixed classification structure of the database – as, e.g., all library systems or search engines like Yahoo – or allow the use of sample documents to search for similar objects in the document collection. Unfortunately, the keyword search approach is not only unusable if the user cannot provide significant terms describing the search request, but also if his retrieval approach is to analyse or browse the document database to gather information about its content. The same problem applies for the content based approach, which requires the availability of a sample document. The fixed classification hierarchy gives a user hints about the contents of a document database and can also support him in browsing. However, a fixed hierarchy requires the user to be familiar with its structure, neglecting the desire of most people to use a structure that reflects their needs and interests. If the user is selecting a wrong path during browsing, because he expects the documents to be classified in this category, he might never find the document(s) he is looking for. Additional support is especially important for users that want to get an overview of a new collection, or for users who are searching for information that is only at abstract levels of a specific topic, which moreover frequently can be described only very vague.

For these reasons, we develop user adaptive methods that can be integrated in an information retrieval tool. The methods should support a user in searching for specific documents as well as analysing and browsing document databases. The currently available software prototype provides, besides standard keyword search methods and the belonging ranked result lists, visualizations of the considered document collection and the search results. For the visualization of the document collection a clustering approach is used, motivated by a self-organizing neural network. During the training process, the content of the database is used. Furthermore, during the retrieval process user feedback is used to interactively adapt the visualization and of search results with respect to user specific needs that might also change over time.

In this article we present some aspects of our work on this topic. We will motivate and discuss the underlying methods of the currently available retrieval prototype, present some applications to text and image databases and give a brief overview of future work.

ANALYZING TEXT DOCUMENT COLLECTIONS

For searching and navigating in large document collections (except of e.g. simple keyword searches) it is necessary to pre-process the documents and store the information in a data structure, which is more appropriate for further processing than an unstructured text file. Despite of its simple data structure without using any explicit semantic information, the vector space model [11] enables very efficient analysis of huge document collections and is therefore still used in most of the currently available document retrieval systems. In the following we briefly describe the vector space model and corresponding document encoding techniques.

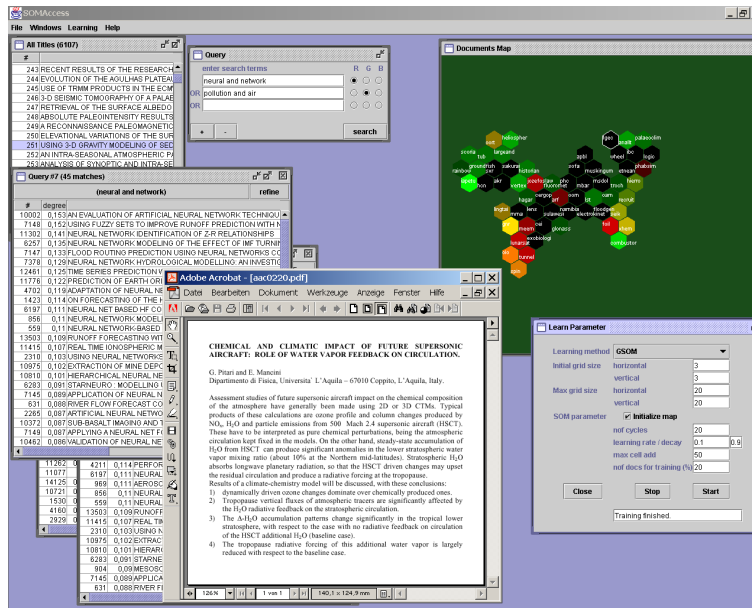


Figure 1. Screenshot of a text document retrieval prototype

The Vector Space Model

The vector space model represents documents as vectors in a t -dimensional space, i.e. each document i is described by a numerical feature vector $D_i = \{x_1, \dots, x_t\}$. Thus, documents can be compared by use of simple vector [11].

The main problem of the vector space representation of documents is to find an appropriate encoding of the feature vector. The simplest way of document encoding is to use binary term vectors, i.e. a vector element is set to one if the corresponding word is used in the document and to zero if the word is not. This encoding will result in a simple Boolean search if a query is encoded in a vector. Using Boolean encoding the importance of all terms for a specific query or comparison is considered as similar. To improve the performance usually term weighting schemes are used, where the weights reflect the importance of a word in a specific document of the considered collection. Large weights are assigned to terms that are used frequently in relevant documents but rarely in the whole document collection. In [9] a weighting scheme was proposed that has meanwhile proven its usability in practice. Besides term frequency and inverse document frequency, a length normalization factor is used to ensure that all documents have equal chances of being retrieved independent of their lengths.

Based on a weighting scheme a document i is defined by a vector of term weights $D_i = \{w_{i1}, \dots, w_{ik}\}$ and the similarity S of two documents (or the similarity of a document and a query vector) can be computed based on the inner product of the vectors, i.e.

$$S(D_i, D_j) = \sum_{k=1}^m w_{ik} \cdot w_{jk} .$$

Thus, this model fulfills a prerequisite for the usability in self-organizing maps. For a more detailed discussion of the vector space model and weighting schemes see, e.g. [1, 10].

Visualization

Visualization of document collections requires methods that are able to group documents based on their similarity and furthermore that visualize the similarity between discovered groups of documents. Self-organizing maps are an approach which is frequently used in data analysis to cluster high dimensional data and which considers also the similarities of neighbouring clusters [4]. This means that not only objects that are assigned to one cluster are similar to each other, but also objects of nearby clusters are expected to be more similar than objects in more distant clusters. Therefore, self-organizing maps can be used to arrange documents based on their similarity and thus to support a user in browsing through a document collection (e.g. [2, 5]).

We have chosen a growing self-organizing map approach to visualize document collections [6], since this variation adapts the size of the map to the complexity of the document collection. The growing self-organizing map can be trained iteratively by the documents the user is using, e.g. based on his email traffic or based on search results. Therefore, the visualization adapts to the documents the user is accessing. Furthermore, it implicitly visualizes changes in the considered data collections [7]. A screenshot of the implemented approach in a retrieval system for text documents is shown in Figure 1. The retrieval tool supports keyword as well as prototype based searching. This approach opens up several appealing navigation possibilities. If the user is looking for objects similar to a given sample object (associative

search), the sample object can be directly mapped to the document map. The grid cell that is selected as winner unit refers to the objects that are most similar to the provided sample. The user can also use this winning unit as a starting point for navigating to similar objects in its neighbourhood. To improve the visualization, the map can be coloured with respect to the distance to surrounding grid cells [3], giving the user hints about the structure of the underlying database. Furthermore, keyword search results can be visualized by coloring the grid cells according to the hits for specific keywords or keyword combinations [6].

INCORPORATING USER FEEDBACK

The self-organizing map performs the clustering of the objects in an unsupervised manner. Therefore it depends on the choice of the description vector and of the definition of similarity between them. This is especially true for multimedia data, where a general definition of significant features is rather hard and strongly depends on the application. However, the user has often a good intuition of a ‘correct’ clustering. Therefore it seems to be very important to incorporate user feedback information to optimise the classification behaviour by gathering information about the desired similarity measure and hereby modifying, e.g., the importance of the selected features using weights. To allow the user to give feedback information so that the visualization can be adapted, the tool was extended in such a way that a user can drag one or several objects from one node of the map to another that in his opinion is more appropriate for the considered objects. Furthermore, the user can mark objects that should remain at a specific node, thus preventing the algorithm from moving them together with the moved object after re-computing the groups on the map. To be able to remap documents we have to change the underlying similarity measure and/or the prototypes that are assigned to specific grid cells. In [8] we proposed user-feedback models for an image retrieval prototype, which we describe briefly in the following.

Learning a user specific weighting scheme

To be able to learn a user specific similarity measure, we replaced the Euclidean similarity function used for the computation of the winner nodes by a weighted similarity measure. Therefore, the distance of a given feature vector to the feature vectors of the prototypes is computed by

$$e_s = \left(\sum_i w^i \cdot (x_s^i - y_k^i)^2 \right)^{\frac{1}{2}},$$

where w^i is a weight vector, y_k the feature vector of an object k and x_s the prototypical feature vector assigned to node s . We have implemented two approaches to compute a weighting scheme based on the users feedback: A global and a local weighting scheme. Both approaches have in common that they iteratively change the weights until the dragged images are mapped to the nodes defined by the user. This is done by increasing the weights for similar features in the document and the target node, and decreasing the weights for similar features of the document and its current node and vice versa decrease and increase the weights of dissimilar features.

The global weighting scheme, which computes a single weight vector for all nodes, can not reflect features that are more important for a local identification of image groups. The global weighting scheme emphasizes more general characteristics that support a good overall grouping of the data collection. This may lead to groups of map neurons to which quite similar images are assigned. Here some features – which are of less importance on a global scope – might be more important to distinguish between local characteristics. Therefore, the second approach assigns individual weights to each node and modifies only the weights of the target and source nodes.

Obviously, this weighting approach also affects the assignments of all other documents. This is intended: The idea of our approach is to interactively find a feature weighting scheme that improves the overall classification performance of the map. Without a feature weighting approach the map considers all features equally important. The global weights can also be used to distinguish features that the user considers important or less important. If, for example, text documents are used where the features represents terms, then we might get some information about keywords the user seem to consider as important for the classification of documents.

Modifying the cluster prototypes: semi-supervised clustering

The idea of this approach is to fine-tune or re-compute parts of the self-organizing map to guide the learning process in how the high-dimensional feature space should be folded (i.e. non-linearly projected) into the two dimensions of the map. In order to modify the map such that it reflects the user changes appropriately, we modified the self-organizing map algorithm so that it considers that an object should be assigned to a specific group (or class). The semi-supervised learning was realized by ignoring the real distances between moved (or fixed) image and prototypes when the winner node is determined. Instead, the target node is always selected as winner for the moved image. Thus, the algorithm moves the prototype of the target node slightly towards the moved images.

INTEGRATION IN A TEXT RETRIEVAL PROTOTYPE

Meanwhile, we have integrated a similar method in our text document retrieval tool, such that a user can adapt the visualization so that the documents are mapped – if possible – according to his needs. The results are similar to the results of the image retrieval prototype discussed in [8]. Furthermore, by interpreting the learned weighting scheme, it is possible to get some information about the user interests. E.g., if the weight of a keyword is strongly increased during the adaptation process, we can expect that the users wants to separate/sort the documents with respect to this keyword and thus this keyword seems to be of special importance for him. On the other hand, keywords that are of little relevance – concerning the clustering – get small weights.

CONCLUSIONS

Self-organizing maps provide valuable means for visualization and exploration of any object collection that can be described by numerical vectors. The combination with keyword search (on, e.g., image annotations) and colouring methods allows the design of interactive and user-friendly object retrieval tools. The presented approaches incorporate user-feedback refining the map that was initially trained in an unsupervised manner by modifying the underlying similarity measure and thus considering user specific grouping criteria.

In future research we will further refine the computation of the weighting scheme, for example, by trying to combine the described methods in such a way that the prototypes assigned to the map nodes as well as the global and local weights are synchronously adapted. Furthermore, we will apply this approach to text databases to analyze its usability in different domains.

ACKNOWLEDGEMENTS

The work presented in this article was partially supported by BTextact Technologies, Adastral Park, Martlesham, UK.

REFERENCES

- [1] R. Baeza-Yates, and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison Wesley Longman, 1999.
- [2] T. Honkela, Self-Organizing Maps in Natural Language Processing, Helsinki University of Technology, Neural Networks Research Center, Espoo, Finland, 1997.
- [3] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, *Newsgroup Exploration with the WEBSOM Method and Browsing Interface, Technical Report*, Helsinki University, Neural Networks Research Center, Espoo, Finland, 1996.
- [4] T. Kohonen, Self-Organized Formation of Topologically Correct Feature Maps, *Biological Cybernetics*, 43, pp. 59-69, 1982.
- [5] X. Lin, G. Marchionini, and D. Soergel, A selforganizing semantic map for information retrieval, In: *Proc. of the 14th International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pp. 262-269, ACM Press, New York, 1991.
- [6] A. Nürnberger, Interactive Text Retrieval Supported by Growing Self-Organizing Maps, In: T. Ojala (edt.), *Proc. of the International Workshop on Information Retrieval (IR'2001)*, pp. 61-70, Infotech, Oulu, Finland, 2001.
- [7] A. Nürnberger, and M. Detyniecki, Visualizing Changes in Data Collections Using Growing Self-Organizing Maps, In: *Proc. of International Joint Conference on Neural Networks (IJCNN 2002)*, pp. 1912-1917, IEEE, Piscataway, 2002.
- [8] A. Nürnberger, and A. Klose, Improving Clustering and Visualization of Multimedia Data Using Interactive User Feedback, In: *Proc. of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002)*, 2002.
- [9] G. Salton, J. Allan, and C. Buckley, Automatic structuring and retrieval of large text files, *Communications of the ACM*, 37(2), pp. 97-108, 1994.
- [10] G. Salton, and C. Buckley, Term Weighting Approaches in Automatic Text Retrieval, *Information Processing & Management*, 24(5), pp. 513-523, 1988.
- [11] G. Salton, A. Wong, and C. S. Yang, A vector space model for automatic indexing, *Communications of the ACM*, 18(11), pp. 613-620, 1975, (see also TR74-218, Cornell University, NY, USA).

