A SOM-Based Technique for a User-Centric Content Extraction and Classification of Web 2.0 with a Special Consideration of Security Aspects

Amirreza Tahamtan[1], Amin Anjomshoaa[1], Edgar Weippl[1,2], and A. Min Tjoa[1]

[1] Vienna University of Technology, Dept. of Software Technology & Interactive Systems, Information & Software Engineering Group
{tahamtan,anjomshoaa,amin}@ifs.tuwien.ac.at
[2] Secure Business Austria, Favoritenstrae 16 - 2nd floor, 1040 Vienna, Austria
eweippl@securityresearch.at

Abstract. Web 2.0 is much more than adding a nice facade to old web applications rather it is a new way of thinking about software architecture of Rich Internet Applications (RIA). In comparison to traditional web applications, the application logic of modern Web 2.0 applications tends to push the interactive user interface tasks to the client side. The client components on the other hand negotiate with remote services that deal with user events. The user should be assisted in different scenarios in order to use the existing platforms, share the resources with other users and improve his security. In this paper we present a user-centered content extraction and classification method based on self-organizing maps (SOM) as well as a prototype for provided content on Web 2.0. The extracted and classified data serves as a basis for above mentioned scenarios.

Keywords: Web 2.0, Self-organizing maps, Extraction, Classification, Security.

1 Introduction

Web 2.0 makes a better use of the client and at the same time pushes the SOA paradigm [23] to its limits. Web 2.0 envisions building collective intelligence and mashed up functionality [4] based on web services. Web 2.0 Users should be supported in three different scenarios:

Assistive services: The specific Web 2.0 contents should provide assistance for users who create similar contents. By analyzing existing contents, some templates and structures should be established and suggested to other users for common contents. It is important to note that the assistive services are not allowed to share sensitive data with other users.

Resource sharing: The data sharing on Web 2.0 is decided by the content owner and there is no holistic solution to avoid an unwanted information disclosure. There is an ever growing need for intelligent sharing of information based on the context.

Y. Bi and M.-A. Williams (Eds.): KSEM 2010, LNAI 6291, pp. 412–423, 2010.

Self-monitoring of trust level: The data contributed by users on Web 2.0 is a source of judgement about individual/organizational behaviors and attitude. This includes the membership in social networks, user groups, contributions on Wikipedia, blog entries, shared videos and pictures and virtual games. In some cases these inferences are not correct and the individuals and organizations have no means to prevent false judgements.

Whatever the intended scenario, the Web 2.0 solutions should satisfy the following requirements:

- The platform should be generic and scalable. Scalability is important since the information on the web is being rapidly increased and the platform should cope with huge amount of data.
- Definition of standard methods for analysis of Web 2.0 documents according to specific resource data models. The result of content analysis should provide the feed to data sharing policies.
- Supporting the user for the development of Web 2.0 content by formal identification of relevant information by means of content analysis results, user behavior and domain ontology.
- Relating the web items that are published in different languages and mapping them to the same ontology in order to make better inferences.

The prerequisite for reaching the above mentioned goals of assisting the user, is automatic extraction and classification of data. For example if a user wants to use available templates on the Web for brainstorming, already created templates should be extracted, ranked and subsequently suggested to the user, as e.g. proposed in [5]. In other words, available templates must be extracted and classified. In another scenario if a user wants to check and monitor his contributions on the Web, e.g. for movie rating sites his entries should be extracted and classified according to the topic. Because of the wide range of platforms on the Web 2.0 from social networks to movie rating sites, performing such tasks manually is very tedious and almost impossible.

This paper provides an overview on the overall approach of our project Secure 2.0 (Securing the Information Sharing on Web) 2.0 [3]. The main contribution of this paper is the introduction of an approach and prototype based on self-organizing maps (SOM) that extracts and classifies the provided content on Web 2.0. The results can be used for assisting the users in the above mentioned scenarios and serves as a basis for these aims. The presented approach is user-centered in the sense that the user himself can assess his web presence before being evaluated by other persons or authorities.

2 Security on Web 2.0

The need for usable and trusted privacy and security is a critical area in the management of Web 2.0 information. This goal demands not only efficient security and privacy policies but also requires improvements of the usability of security aspects.

The disclosure of personal/organizational information on Web 2.0 has created new security and privacy challenges. Designing transparent, usable systems in support of personal privacy, security and trust includes everything from understanding the intended use of a Web 2.0 system to users' tasks and goals as well as the contexts in which the users use the system for information sharing purposes.

Web 2.0 security concerns are divided into two basic categories: physical security and semantic security. The former aims to cover issues such as secure and trustworthy data exchange. This group of security concerns can benefit from existing methodologies of Web 1.0. The semantic security handles the information sharing on a higher level by exploiting the Semantic Web technologies in order to describe the shared knowledge in a computer-processable way. As a result the shared information can be combined with personal/organizational policies to protect the information in a collaborative environment like Web 2.0.

Obviously new security and privacy schemas are required to cover the requirements of Web 2.0 applications which are being raised due to the following reasons:

- Web services are the building blocks of Web 2.0 applications and by liberating web services from organizational environments, it is necessary to have appropriate information disclosure and information usage policies.
- Web 2.0 has made the content creation much easier and as a result a huge amount of data is constantly created. The volume of data on the web is doubled since the emergence of Web 2.0 technologies. Data mining of user generated entities and extracting knowledge and information patterns is a new threat to privacy of individuals.
- The Web 2.0 architecture is tending to utilize the client side processing power. Hence, Web 2.0 can be used intelligently for better integration of user data with global business processes. In other words the "user desktop" can interact with the real world processes and provide the requested data without human interaction. Before such dreams can come true, we need an efficient mechanism to define users' security and privacy policies.

3 Self-Organizing Maps

Self-organizing maps (SOM) or self-organizing feature maps, also sometimes called Kohonen maps, have been introduced by Teuvo Kohonen in [12]. SOM belongs to the family of artificial neural networks and uses unsupervised learning. It can be used for reducing the dimensionality of the input by mapping the input onto a low-dimensional (usually two dimensional) grid. Each input object is represented as an N-dimensional vector. Each dimension of the vector is called a feature. Each node in the grid is assigned a weight vector which is again represented by an N-dimensional vector. Components of a weight vector are assigned at initialization time random values. The SOM's learning process is as follows: a node in the gird with the minimum distance to a given input is chosen. This node is called the winning node. The components of the weight vector of the

winning node are adjusted such that the distance with the input vector becomes smaller. The components of the weight vectors of the neighboring nodes of the winning node are as well adjusted. This cycle is iterated until it converges, i.e. no more adjustments are performed. Figure 1 depicts an architecture based on SOM for information retrieval from documents. For more details on analyzing textual data we refer to subsection 3.1 and section 4. The main characteristic of SOM is preserving topology, i.e. the neighborhood and distance relationship between input data is preserved and by mapping becomes explicit. SOM maps more frequent input data onto lager domains compared to less frequent input data. Several Authors have proposed different variants of SOM and several algorithm for it has been developed, e.g. [9,8].

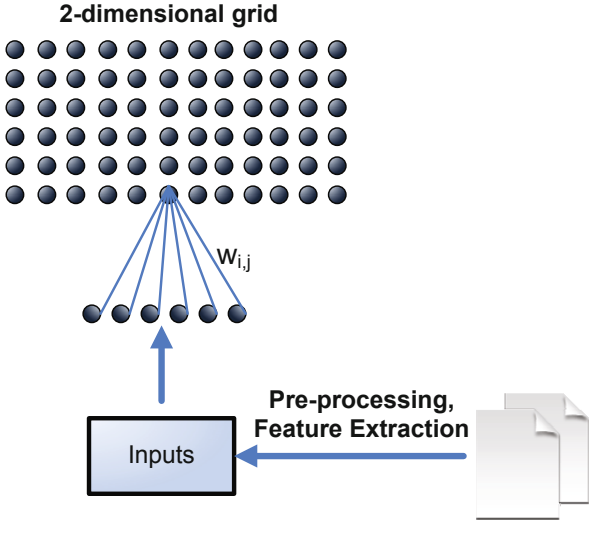

Fig. 1. A SOM-Based approach for information retrieval from documents

SOM has many applications in different domains. An overview on SOM-related literature can be found in [10,18].

3.1 SOM-Based Text Analysis

In this work we use SOM for automatic clustering of high-dimensional data. The SOMs can be visualized and the distance between concepts depicts their similarity with regards to some predefined features. A typical SOM algorithm for classification of text based items can be summarized as follows [6]:

1. Initialize input nodes, output nodes, and weights: Use the top (most frequently occurring) N terms as the input vector and create a two-dimensional map (grid) of M output nodes. Initialize weights w_{ij} from N input nodes to M output nodes to small random values.

2. Present each document in order: Describe each document as an input vector of N coordinates. Set a coordinate to 1 if the document has the corresponding term and to 0 if there is no such a term.
3. Compute distance to all nodes: Compute Euclidean distance d_j between the input vector and each output node j:
   $d_j = \sum_{i=0}^{N-1} (x_i(t) - w_{ij}(t))^2$
   where $x_i(t)$ can be 1 or 0 depending on the presence of i-th term in the document presented at time t. Here, w_{ij} is the vector representing position of the map node j in the document vector space. From a neural net perspective, it can also be interpreted as the weight from input node i to the output node j.
4. Select winning node j^* and update weights to node j^* and its neighbors: Select winning node j^*, which produces minimum d_j. Update weights to nodes j^* and its neighbors to reduce the distances between them and the input vector $x_i(t)$:

   $w_{ij}(t+1) = w_{ij}(t) + \eta(t)(x_i(t+1) - w_{ij}(t))$

   After such updates, nodes in the neighborhood of j^* become more similar to the input vector $x_i(t)$. Here, $\eta(t)$ is an error-adjusting coefficient ($0 < \eta(t) < 1$) that decreases over time.
5. After the network is trained through repeated presentations of all documents, assign a term to each output node by choosing the one corresponding to the largest weight (winning term). Neighboring nodes which contain the same winning terms are merged to form a concept/topic region (group). Similarly, submit each document as input to the trained network again and assign it to a particular concept in the map.

Figure 2 shows an example of a self-organizing map with clusters and sub-clusters.

4 The Proposed Approach

There are a handful of techniques and algorithms for text analysis and information retrieval (IR) purposes. It goes without saying that the rudimentary methods such as removal of stop words and stemming are not enough.

To analyze textual data with a self-organizing map, the following steps must be performed:

Data Extraction: The first step toward using Web 2.0 contents is to analyze and extract that data. In this context the Web 2.0 API of the systems under study will provide the required feeds for the text analysis component.

Pre-processing: This step includes applying stemming algorithms, removing stop words, format conversion, etc.

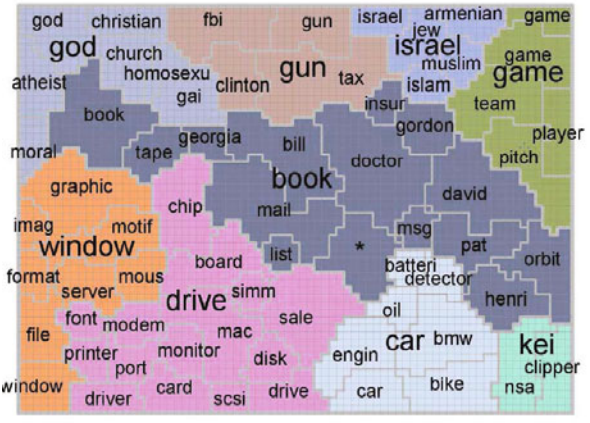

Fig. 2. Self-organizing map with cluster and sub-clusters, image from [22]

Feature Extraction: Feature extraction refers to description of data characteristics. Note that self-organizing maps are only capable of handling numerical data. Hence features need to be presented in a numerical form. Textual data can be represented by different approaches such as bag-of-words [28], phrase detection [11] and Latent Semantic Indexing [7]. We use a binary presentation of features, i.e. components of the input vectors, which represent the words contained in the document are assigned 1 if this word is contained in the document and 0 otherwise. The features of a document will be presence or absence of words in this document. For example consider the following entries on Twitter. Twit 1: "The Guardian newspaper has announced it will support the Liberal Democrats" and Twit 2:"A woman brought you into this world, so you have no right to disrespect one". Their extracted vector is depicted in the figure 3. The vector shows the results after the pre-processing step, e.g. removal of stop words, stemming, etc. That is why only a subset of the words are included in the vector. Note that this a simplified version. In our experiments we consider the correct sense of the word in the context using the WordNet [16] and augment the vector with this information as described in subsection 3.1.

Training: After the input data has been prepared, it can be used to train the self-organizing maps. There is no general rule about the map size. However, the number of data items must be sufficiently large enough compared to the map size. According to [13], eleven training samples per node is just about sufficient.

Visualization: The maps can be visualized using a variety of methods such as: Vector Activity Histogram, Class Visualization [15], Component Planes [27], Vector Fields [20], Hit Histogram, Metro Map [17], Minimum Spanning Tree, Neighborhood Graph [21], Smoothed Data Histograms [19], Sky Metaphor Visualization [14], U-Matrix [26], D-Matrix, P-Matrix [24] and U^*-Matrix [25].

	guardian	newspaper	announce	support	liberal	democrat	woman	bring	world	right	disrespect
Twit 1:	1	1	1	1	1	1	0	0	0	0	0
Twit 2:	0	0	0	0	0	0	1	1	1	1	1

Fig. 3. An example of an input vector

Furthermore as an additional task we have to take the extra information of Web 2.0 entries into consideration throughout the document processing phase. As mentioned before, the Web 2.0 entries might have an additional semi-structured information such as tags, attachments, relations, etc. This information can play an important role in the item analysis of Web 2.0 entries and should be extracted and included in the input vector of the SOM algorithm. At the end of the SOM process we end up with a group of data points that are merged together and form clusters. The clusters are labeled and this labeling might be further improved in the next step for creating the ontologies. An important matter in this step is consideration of security and privacy issues. After the analysis of documents the sensitive content of the nodes should be removed. The use of the extracted ontology together with context and security ontology make this possible. Security issues are discussed in more detail in subsection 4.1.

4.1 Security and Privacy Solutions

The text analysis process of the previous step is focused on a domain-independent, statistical analysis of the text. This should be combined with context information. The statistically derived "implicit semantics" of Web 2.0 entries should be annotated and aligned with "explicit semantics" which is based on formal ontology and domain knowledge. Ontology and semantic metadata also play a critical role in combining the existing knowledge with application context by scoring and ranking the fitting candidate information. Moreover the "explicit semantics" is used to bridge the gap between content, users and policies via their relevant ontologies. The "implicit semantics" of our SOM-approach may play the role of an indicator to specify the priority of required domain ontologies. As a result the Web 2.0 items plus domain ontology provide an elaborated set of information for improving inference and query answering processes. Ontologies are used as the basis for providing assistive-services and information sharing. Also in self-monitoring use case the ontology plays a crucial role to bind the information from different resources together.

Part of the alignment process can be done automatically by selecting the appropriate ontology from available domain ontologies and annotating the Web 2.0 items with ontology concepts. The major applications of ontology alignment are as follows:

– Formal domain ontology together with security and privacy ontologies make it possible to identify and anonymize the sensitive data. The result can be safely shared with other users or be reused via an assistive service as a template.

– Domain ontologies can be also combined with user policy to restrict the information sharing and apply necessary filters.
– Ontology alignment process can be used as an input for ontology engineers to enrich the domain ontologies based on common data structures.

The Web 2.0 items via their relevant domain ontologies is connected to other information resources for more elaborated tasks such as reasoning or complex queries. As soon as we arrive in the ontology level, the solutions can benefit from the concrete research works about semantic security and privacy. Figure 4 depicts the overall solution and the important role that ontologies play in connecting the distributed knowledge domains

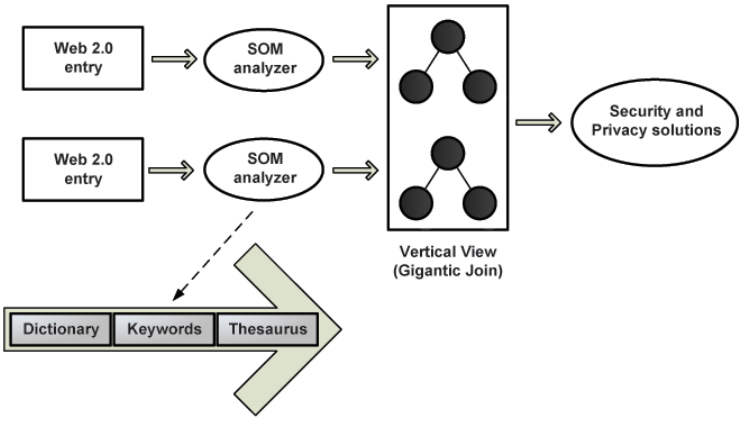

Fig. 4. overall solution and ontological join of knowledge domains

5 Experiments and Prototypical Implementation

Before starting with knowledge extraction and advanced text processing techniques we setup a data resource for our experiments. For this purpose three major Web 2.0 platforms: YouTube, Flickr, and Twitter, plus MindMeister [1] were selected as the basic data resources. Java components that use the corresponding REST APIs of these platforms for extracting data items and storing them locally is implemented. Figure 5 shows a screen shot of this component. The extracted data can be again used as services. It is important to note that the extractor also plays an important role in the runtime system and will be used to create temporary data sources for user-generated scenarios on the fly. Furthermore a prototype for disambiguation and annotation of mind map content based on WordNet dictionary is implemented. The disambiguation results are then used to annotate the text with the correct sense of the word which is necessary before starting up with SOM for getting better results, i.e. to improve the quality of SOMs and decrease the error function in terms of misclassification. After feature extraction from documents, the SOM analysis is performed using the Java SOMToolbox [2].

Fig. 5. Data extractor and feed components

Figure 6 illustrates an experiment on Twitter entries. This experiment includes 1754 twits (number of vectors) and 1282 words after removal of stop words (number of features) on a 11×11 map. Due to visibility reasons only a part of the map is visualized here. The numbers on the map show how many twits are about the same topic and therefore are mapped onto the same region. The black circles identify the twits that maybe problematic or about inappropriate topics. Using this approach the user can identify that a person he is following on the twitter is posting entries about topics which he does not want to be linked with, e.g. hate literature, and the user may want to remove him from his list. Identification of out of favor topics or entries are user's responsibility. For example employee of a company may not want to be linked to positive posts on competitors' products whilst other users have no problem with that. As users have different legal, ethical, personal, religious and job ethics, we have left this decision up to the user.

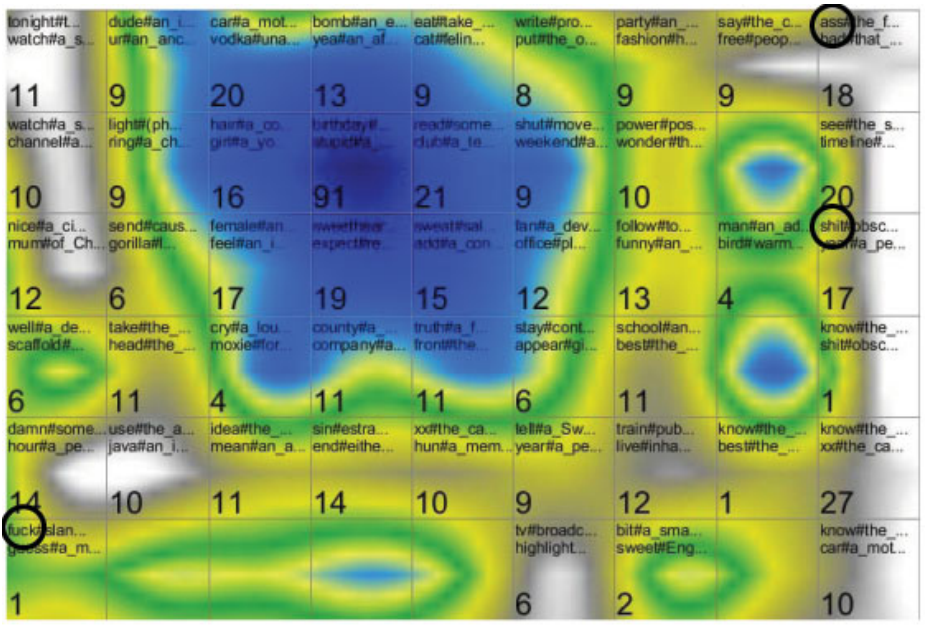

Fig. 6. Smoothed data histogram visualization of Twitter entries

6 Conclusions and Future Work

Data extraction and classification can be used to in many ways to assist the user in different scenarios. In this paper we have presented a generic approach for extraction and classification of data in a user-centered way. In other words, the user himself can monitor his web presence before being evaluated by others. We use SOMs for classification and categorization of textual data. The provided results are parts of the ongoing project secure 2.0.

We plan to to use Mashups to create situational solutions of our platform. This decision has two main benefits: first of all mashups will enable us to setup and test different approaches and service composition variants. Second, the end user of system who is not an IT expert can easily put the services together and create a customized solution that meets his requirements. There are also some other factors that will be studied and tested. For example the weight selection and the number of clusters are two major parameters in using SOM for clustering purposes. The first experiments show that we need a method to decrease and merge smaller clusters while preserving the topology of data. Inappropriate selection of clusters will result in huge number of clusters that will be computationally intensive. In this regard we are planning to benefit from other clustering methods such as K-means clustering. Another question is the optimal number of iterations in the training phase, the learning rate and the appropriate size of the 2-dimensional grid. Another problem that we are dealing with is the huge

amount of data that SOM should handle. In this regard we examine how to enhance the capabilities of the suggested algorithms for self-organizing maps such as scalable self-organizing maps to cater for the significant amount of data and how we can use the characteristics of the data present on Web 2.0 to reduce the computational complexity.

Acknowledgments. This work is supported by the Austrian FIT-IT project Secure 2.0.

References

1. http://www.mindmeister.com/
2. http://www.ifs.tuwien.ac.at/dm/somtoolbox/index.html
3. Secure 2.0 - securing the information sharing on web 2.0, http://www.ifs.tuwien.ac.at/node/6570
4. Anjomshoaa, A.: Integration of Personal Services into Global Business. PhD thesis, Vienna University of Technology (2009)
5. Anjomshoaa, A., Sao, K.V., Tjoa, A.M., Weippl, E., Hollauf, M.: Context oriented analysis of web 2.0 social network contents - mindmeister use-case. In: Proc. of the 2nd Asian Conference on Intelligent Information and Database Systems (2010)
6. Chen, H., Schuffels, C., Orwig, R.: Internet categorization and search: a machine learning approach. Journal of Visual Communications and Image Representation 7(1), 88–102 (1996)
7. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society For Information Science 41, 391–407 (1990)
8. Fritzke, B.: Growing cell structures–a self-organizing network for unsupervised and supervised learning. Neural Networks 7(9), 1441–1460 (1994)
9. Kangas, J.A., Kohonen, T., Laaksonen, J.T.: Variants of self organizing feature maps. IEEE Transactions on Neural Networks 1(1), 93–99 (1990)
10. Kaski, S., Kangas, J., Kohonen, T.: Bibliography of self-organizing map (som) papers: 1981-1997. Neural Computing Surveys 1(3-4), 1–176 (1998)
11. Kawahara, T., Lee, C.H., Juang, B.H.: Combining key-phrase detection and subword-based verification for flexible speech understanding. In: Proc. of the International Conference on Acoustic, Speech, Signal Processing (1997)
12. Kohonen, T.: The self-organizing map. Proc. IEEE 78, 1464–1480 (1990)
13. Kohonen, T., Somervuo, P.: Self-organizing maps of symbol strings. Neurocomputing 21(1-3), 19–30 (1998)
14. Latif, K., Mayer, R.: Sky-metaphor visualisation for self-organising maps. In: Proc. of the 7th International Conference on Knowledge Management (2007)
15. Merkl, D., Rauber, A.: Alternative ways for cluster visualization in self-organizing maps. In: Proc. of the Workshop on Self-Organizing Maps (1997)
16. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM 38(11), 39–41 (1995)
17. Neumayer, R., Mayer, R., Rauber, A.: Component selection for the metro visualisation of the som. In: Proc. of the 6th International Workshop on Self-Organizing Maps (2007)
18. Oja, M., Kaski, S., Kohonen, T.: Bibliography of self-organizing map (som) papers: 1998-2001 addendum. Neural Computing Surveys 3, 1–156 (2002)

19. Pampalk, E., Rauber, A., Merkl, D.: Using smoothed data histograms for cluster visualization in self-organizing maps. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, p. 871. Springer, Heidelberg (2002)
20. Poelzlbauer, G., Dittenbach, M., Rauber, A.: Advanced visualization of self-organizing maps with vector fields. Neural Networks 19(6-7), 911–922 (2006)
21. Poelzlbauer, G., Rauber, A., Dittenbach, M.: Advanced visualization techniques for self-organizing maps with graph-based methods. In: Proc. of the Second International Symposium on Neural Networks (2005)
22. Roiger, A.: Analyzing, labeling and interacting with soms for knowledge management. Master's thesis, Vienna University of Technology (2007)
23. Tahamtan, A.: Modeling and Verification of Web Service Composition Based Interorganizational Workflows. PhD thesis, University of Vienna (2009)
24. Ultsch, A.: Maps for the visualization of high-dimensional data spaces. In: Proc. of the Workshop on Self-Organizing Maps (2003)
25. Ultsch, A.: U*-matrix: a tool to visualize clusters in high dimensional data. Technical Report Technical Report No. 36, Dept. of Mathematics and Computer Science, University of Marburg, Germany (2003)
26. Ultsch, A., Siemon, H.P.: Kohonen's self-organizing feature maps for exploratory data analysis. In: Proc. of the International Neural Network Conference (1990)
27. Vesanto, J., Ahola, J.: Hunting for correlations in data using the self-organizing map. In: Proc. of the International ICSC Congress on Computational Intelligence Methods and Applications (1999)
28. Wallach, H.M.: Topic modeling; beyond bag of words. In: Procs. of the International Conference on Machine Learning (2006)