

Developing Initial State Fuzzy Cognitive Maps with Self-Organizing Maps

Marcel Wehrle, Edy Portmann, Alexander Denzler, and Andreas Meier

University of Fribourg
Boulevard de Perolles 90, 1700 Fribourg, Switzerland
University of Bern
Engehaldenstrasse 8, 3012 Bern, Switzerland
{marcel.wehrle, alex.denzler, andreas.meier}@unifr.ch
edy.portmann@iwi.unibe.ch

Abstract. Using soft computing methods, the authors collect and process relevant user-generated information from the web. Through the use of self-organizing maps, fuzzy cognitive maps are constructed. The fuzzy cognitive map is a generated representation of the emergent web semantics of the dataset. In the next step, the fuzzy cognitive maps are enriched with related lexical content and stored in a graph database. This makes it possible for a human user to explore the maps in a visual way. Following a design science research approach, a prototype has been implemented as a proof of concept.

Keywords: Emergent web semantics, Fuzzy cognitive maps, Self-organizing maps

1 Introduction

Today's existing social web contains a set of relations that connect users through the Internet [1]. It primarily consists of human-understandable information (i.e., semantics); however, this information most often comes in unstructured form and is thus not straightforward to interpret by computers. However, the semantic web should enable computers to understand and respond to complex user queries based on their meaning [2]. Such an understanding, however, requires relevant information (i.e., data that have been given formal meaning by way of relational connection through some meaning negotiation process [3]) to be semantically structured. To this end, emergent web semantics is a possible answer to enhance the interaction between humans and machines.

This field consists of a set of methods and techniques for analysing the evolution of decentralized semantic structures in large-scale distributed information systems [4]. As inevitably required in the semantic web (i.e., to adaptively and dynamically address today's information explosion as naturally as possible), emergent semantics adopts a complex systems approach of addressing meaning by automatically creating semantics in a distributed system as an ensemble of relationships between syntactic structures [5]. Both the discovery of the proper

interpretation of symbols (e.g., as a result of a self-organizing process performed by distributed agents) and the representation of the thus created semantics are taken into account.

To team up humans and computers, both need the ability to learn from each other. This may progress quite naturally because over time, our ability to process information and communicate it to others improves [6]. For example, through dynamic query interfaces [7], humans are able to adapt to computers, and through machine learning, computers can (better) adapt to humans. This computational intelligence consists of a toolbox of nature-inspired methods of computation to address the real world's complexity to which conventional approaches (i.e., first principles modelling or statistical modelling) are ineffectual or impracticable [8]. It also embraces biologically-inspired algorithms (i.e., swarm intelligence and artificial immune systems), which can be seen as part of evolutionary computation, and includes broader fields such as natural language processing and data mining.

To support people in their searches, dynamic query interfaces should fit into a single user's knowledge. For this purpose, the computer should rely on learning to defer to the average user, who uses language in a natural way. Through dynamic interfaces that integrate digital content into a human's life in seamless ways, a computer should even become adaptable to each individual user. With an automatically built-in knowledge graph (a fuzzy cognitive map (FCM)), computers may become (more) responsive to humans. These FCMs may be created automatically on the basis of self-organizing data mining algorithms (i.e., agents that crawl and aggregate social web data). A similar approach was already introduced by Jazzar & Jatan [9] by using FCMs in SOM-based intrusion systems. Normally, FCMs are constructed out of human knowledge [10]; consequently, they strongly depend on the subjective beliefs of the expert(s). Furthermore, the map itself is limited to be relatively simple and small and domain specific [11]. To overcome this, several efforts have been conducted by introducing algorithms to learn the FCM model structure. In general, there are two main proposed paradigms: Hebbian and genetic algorithms [12–16]. However, with the help of SOM, the aim of this framework is to find an initial state vector, which is constructed out of the collected data, that leads to a predefined FCM, similar to Kahn & Chong [17].

Section 2 introduces fuzzy logic, cognitive maps and FCMs. Section 3 describes self-organizing maps (SOMs), their functionality and how they can be used to construct initial FCMs. Based on SOMs, Section 4 presents graph databases as possible stores and query engines for emergent semantics [4]. In Section 5, the authors' framework is presented, and its architecture and single components are illustrated. Section 6 presents the prototype with all components. Finally, in Section 7, we draw conclusions and indicate possibilities for future research.

2 Fuzzy Cognitive Maps

The problem with today's semantic web is that the more complex it becomes, the less precise the statements (i.e., exact statements formulated in two-valued predicate logic) that can be made about it become. This is the message of the

principle of incompatibility, which roughly states that high complexity is incompatible with high precision [18]. Therefore, anticipating an enhancement to two-valued logic, Zadeh introduced fuzzy logic as a tool for formalizing and representing the reasoning process and fuzzy logic systems, which are based on fuzzy logic and possess many characteristics attributed to intelligence [19]. Fuzzy logic effectively handles uncertainty, which is common in human reasoning, perception and inference and, contrary to some misconceptions, has a very formal and strict mathematical backbone (i.e., it is deterministic in itself, yet it allows uncertainties to be effectively represented and manipulated).

In more detail, fuzzy sets are graduated in the sense that membership in a fuzzy set is a matter of degree. A fuzzy set A , in a universe of discourse U , is defined by a membership function u_A that associates with each object u in U , the degree to which u is a member of A . A fuzzy set is basic if its membership function takes values in the unit interval $[0, 1]$. More generally, the membership function may take values in a partially ordered set.

FCMs are fuzzy structures that strongly resemble neural networks, and they have powerful and far-reaching consequences as a mathematical tool for modelling complex systems. Kosko [20] introduced FCMs as a fuzzy extension of the cognitive map pioneered by Axelrod [21], who used the map to represent knowledge as an interconnected, directed, bi-level logic graph. The underlying model behind FCMs is simple and effective because it can analyse the data using directed graphs and connection matrices [11].

FCMs are fuzzy signed directed graphs with feedback. The directed edge e_{ij} from causal concept C_i to concept C_j measures how much C_i causes C_j . The time-varying concept function $C_i(t)$ measures the non-negative occurrence of some fuzzy event (e.g., the strength of a sentiment, strategy or historical trend). The edges e_{ij} take values in the fuzzy causal interval $[1, 1]$; $e_{ij} = 0$ indicates no causality, $e_{ij} > 0$ indicates a causal increase, and C_j increases as C_i increases (or C_j decreases as C_i decreases).

Finally, $e_{ij} < 0$ indicates a causal decrease or negative causality, and C_j decreases as C_i increases (and/or C_j increases as C_i decreases). Simple FCMs have edge values of 1, 0, 1. Thus, if causality occurs, it occurs to a maximal positive or negative degree. As a direct consequence, the values provide a quick first approximation to an expert's causal knowledge.

A solution for integrating computer-understandable meaning in today's semantic web is often programmed from above, such as a creator of an ontology makes something and imparts it with his or her intelligence. Another approach uses a more bottom-up, decentralized method; bio-inspired techniques often involve the method of specifying a set of simple rules, a set of simple organisms that adhere to those rules and a method of iteratively applying those rules [1]. Because the FCMs are generated from social web information, only allowing FCM patterns that are stabilized over time, it involves continuous optimization, and its algorithms can be considered global optimization methods.

After several generations of rule application, it is usually the case that some forms of complex behaviour arise. Complexity gets built upon complexity un-

til the end result is something markedly complex and, quite often, completely counterintuitive to what the original rules would be expected to produce. Evolutionary computation uses iterative progress, just like the permanent aggregating of information by web agents yields emergent semantics [4]. This emergent semantics may be managed with FCMs.

A FCM can model the relationships of various factors, depending on what logic the word vectors are constructing (e.g., similarity or semantics). They can have three different characteristics: (1) indicating a positive or negative causality of a relationship, (2) showing the strength of causal relationships with fuzzy values, and (3) dynamic causal links where changes affect concepts [16].

In the following section, we introduce SOMs that help create the emergent semantics underlying FCMs.

3 Generating Fuzzy Cognitive Maps with Self-Organizing Maps

Since the introduction of SOMs by Kohonen in 1982 [22], more than 7700 SOM-related research papers have been published, primarily in the fields of image analysis, speech recognition, signal processing and robotics [23]. A SOM is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map. SOMs use a neighbourhood function to preserve the topological properties of the input space (abstraction). This makes SOMs useful for visualizing low-dimensional views of high-dimensional data, akin to multidimensional scaling [22]. Like most artificial neural networks, SOMs operate in two modes: training and mapping. Training constructs the map using input examples (a competitive process, also called vector quantization), whereas mapping automatically classifies a new input vector.

A SOM consists of components called nodes or neurons i . Associated with each node i is a weight vector w_i of the same dimension as the input data vectors and a position in the map space. The usual arrangement of nodes is a two-dimensional regular spacing in a hexagonal or rectangular grid. To overcome the border effect, spherical grids have been introduced [24, 25]. The SOM describes a mapping from a higher-dimensional input space to a lower-dimensional map space (visualization). The procedure for placing a vector from data space onto the map is to find the node with the closest (i.e., smallest distance metric, such as Euclidean distance) weight vector to the data space vector. In general, in the initialization phase, the number of input units and the topology of the output layer are first determined. However, this is difficult because of the increasing amount of available information. To overcome this limitation, several improvements to SOMs have been introduced, mainly by adaption such as growing grid models [26], hierarchical feature maps [27], growing hierarchical maps [28] or tree-structured maps [29].

Large SOMs exhibit emergent properties. In maps consisting of thousands of nodes, it is possible to perform cluster operations on the map itself [30].

4 Storing Web Semantics in Graph Databases

Graph databases serve as a digital storage medium for graphs. At the same time, they provide users with features and functionalities commonly used in the domain of graph theory. Within the database landscape, they are classified as belonging to the group of non-relational data models [31]. This group of databases is viewed as an extension to traditional relational databases. Four different types of databases are prominent representatives in this group, with key-value stores and big table clones focusing on handling big data sizes and documents and graph databases specializing in complex data [32]. Considering that structured knowledge is viewed as highly interconnected and thus as complex data, this backs the proposal for choosing a graph database for storing from a technical perspective.

Not all graph databases are the same, however. Some are constructed on top of other data models, while others are standalone solutions. Other differences exist based on the purpose and environment that they have been developed for. While web-based solutions focus on maintaining low latency times for queries, others aim at handling large graphs by scaling horizontally, whereas others are specifically designed for processing algorithms as fast as possible by storing the entire graph in memory [33].

Because FCMs are highly interconnected directed graphs, it appears reasonable to represent them in graph databases. This allows optimized queries to the database [31]. Furthermore, most graph databases allow visual exploring of the underlying FCMs through a web interface. This facilitates the interaction between humans and the FCMs. Graph databases are a valuable tool for representing the web semantics of a given dataset. Another advantage lies in the computational possibilities offered by graph algorithms. Computing shortest paths, clusters and recommendations are tasks that graph algorithms are particularly suited [31]. This may allow deeper insights into the FCMs.

5 Architecture

The framework is built upon a 3-layer software architecture. Figure 1 shows the underlying design. As an external data source, semistructured content from the web is accessed through a web crawler, an API or a data dump. The first layer consists of all data processing steps, beginning with the manipulation of the data. Here, various options are possible. The content can be normalized through tokenization, stemming or lemmatizing. If the amount of data should be reduced, stopwords or particular parts of speech (through part-of-speech tagging) can be eliminated. For further manipulation of the data, the concepts (i.e., words or phrases) need to be represented as word vectors (e.g., Mikolov et al. [34] or Maas et al. [35] provide unsupervised vector-based approaches). These word vectors form the input vectors to create the SOM. The topology, which results as an output from the SOM, can already provide an adequate overview of the data. It shows clusters of similar concepts and where the main topics of data are located.

The output of the SOM can be used as an input matrix to generate the FCMs. The matrix already contains the information needed to draw the concepts with the causal relationship among these concepts. For the estimation of the causal relationship strength, depending on the use case, one of the proposed learning algorithms (see section 2) can be used.

The application layer includes a web framework to provide accessibility over a web browser. This facilitates the application being used by a broader audience. Because the framework is built upon a graph database, the only restriction is an available interface to the database. Finally, the user interface is an important component to complete the user experience. This interface should allow querying and exploring the underlying datasets such that the user can enlarge his personal knowledge.

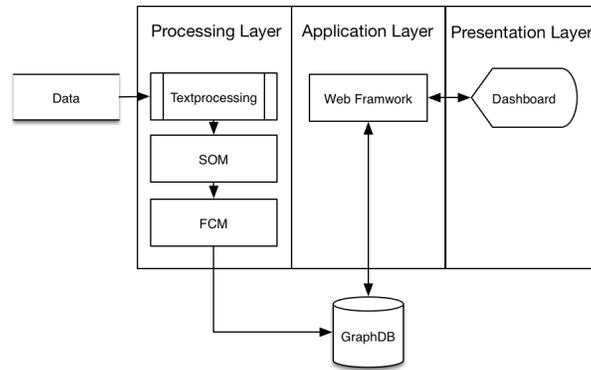


Fig. 1. Framework Architecture

6 Prototype

As a data source, the authors used the dump of a stack exchange on the topic computer science. A primary function of a stack exchange is to share knowledge from user to user. The exchange is performed through questions and answers. Overall, 13492 positively rated posts were considered for the evaluation. Users of the stack had used 463 tags to mark their questions, mostly algorithms (2267), complexity theory (1143) and formal languages (767). Various recent studies have already used the stack dataset [36–38] as a data source, but none considered FCMs.

A framework was established to analyse the data. Followed by the processing steps shown in Figure 2, the data were transformed into FCMs. As soon as the data are normalized and tokenized (1), bigrams are identified and a word vector w_i is constructed as a representation of these bigrams (2). In the prototype,

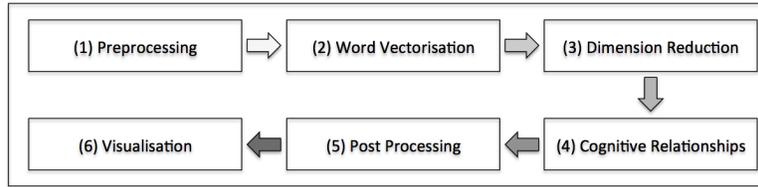


Fig. 2. Processing steps

the authors used a context-prediction vector, CBOW a computationally efficient model by Mikolov [34]. CBOW learns to predict words in the middle of a symmetric window. The window is based on the sum of w of words in the window. However there are various other possibilities to build semantic word vectors [39]. The word vectors w_i are generated for each concept in the vocabulary with a dimensionality of 450.

These vector representations $w = w_1, w_2, \dots, w_n$ are the input source for the SOMs to reduce their dimensionality (3). During the evaluation process, a total of 150 training epochs with a learning rate decreasing from 0.6 to 0.02 have been seen as promising. The size of the map is 225x225 which leads to an output layer of 50625. Figure 3 shows the topology of the explored dataset. The produced distance matrix is the input source for the FCM.

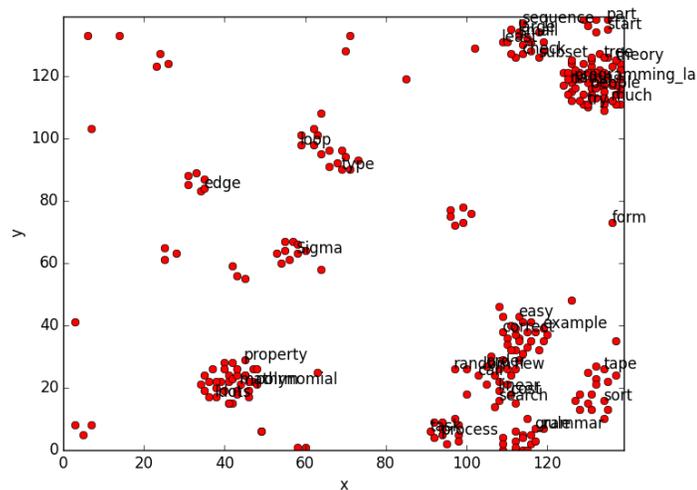


Fig. 3. 2D - Visualization of the SOM of the principal concepts.

The visualization of SOMs also has the advantage of showing centres of concepts where the data are spread. As is often the case in FCMs, weights are determined by human interaction, and learning algorithms are used to eliminate the subjectiveness of weights. In this framework, the weights w_{ij} are adjusted through the closeness of concepts C_j and C_i , $i = 1, 2, \dots, N, j = 1, 2, \dots, N$, in the produced distance matrix of the SOM until a stop criterion is reached. This leads to an initial state vector of the FCM. The setup for the value of this weight is between $[0, 1]$ (4). The related pseudocode is shown in Algorithm 1. Then, the fuzzy expressions medium, strong and very strong are mapped to the numerical values of the relationships (Table 1).

Algorithm 1 FCM algorithm

```

1: procedure CONSTRUCTION FCM
2:    $weight \leftarrow inputDistanceSOM$ 
3:    $nodes \leftarrow concepts$ 
4:    $minDistanceValue$ 
5: loop:
6:   for  $i, j$  in  $nodes$  do
7:     if  $weight(nodes(i, j)) > minDistanceValue$  then
8:        $Construct\ E\{i, j\}$ 
9:        $Label\ E\{i, j\}$ 
10:  close;
```

Algorithm 1 allows the connectivity of the graph to be adjusted by varying the minimal distance value. After the FCM of the dataset is constructed, it is post-processed (5) by adding metadata to each node. These metadata are delivered through an interface to wikidata. All the values, included in the metadata as properties, are stored through a database layer in a graph database. The authors selected OrientDB as a database. OrientDB is a hybrid system that offers a graph database and a document database. It has all the functionality required for this experiment, especially because it possesses a good query system (i.e., it can calculate shortest path operations).

Fuzzy membership functions	Fuzzy regions	Defuzzified value (weight)
medium	(0.6, 0.8)	0.7
strong	(0.7, 0.9)	0.8
very strong	(0.9, 1)	1

Table 1. Fuzzy expression mapping

The final component of the framework is the user interface. OrientDB already provides a complete interface for querying and analysing graph-related data [40].

It also allows direct interactions, so users can explore and manipulate data. In this way, the data can be visualized in a graph-based environment (6). Currently, the produced output of the processed data can be accessed in [41]

7 Conclusion

The authors propose a framework for analysing data from web sources that develop web semantics represented through fuzzy cognitive maps. The framework includes a mixture of different algorithms and technologies. The FCM allows users to explore underlying web sources enriched with metadata. (Hidden) relationships between concepts can be uncovered. The underlying framework also allows direct user interaction to adjust causal relationships in the graph. In this way, a machine-human interaction is established, where both can learn from each other [5]. Note that the prototype only produces the initial state FCM on a limited dataset.

An advantage of this approach is certainly its simplicity. It can be applied on nearly all user-produced content to generate the related FCMs. This already provides a proper overview on which concepts are important and how they are interconnected. FCMs allow the inclusion of uncertainty and vagueness, both inherent in the human language. Thus, a better interpretation of human-produced content may be possible. They also allow a certain flexibility in the setup of the framework. Through the enrichment with metadata, it may also be used to discover knowledge. Limitations are given by the used technologies and inputs. User-generated input is never perfect and often includes spelling or grammatical mistakes. These mistakes could not always be removed. Reducing the dimensionality through SOM simplifies the data, but at the same time, information is lost. Finally, FCMs with many concepts rapidly become unclear and the user may lose focus.

A related framework was already used for a trend discovery project of the Swiss Commission for Technology and Innovation (CTI). In this project, trends, relevant for the tourism industry, have been identified from social media as an input for performance management. The main difference was the temporal component of the framework to view the transformations over time, which allows the change of concepts relating to their relevance and their relationships to be observed. Furthermore, the dataset grows through continuous crawling, and the weights are adjusted through a learning algorithm.

The basic concept forms the basis for the construction of a granular knowledge cube [42]. This concept combines an amplified multi-source knowledge base together with a knowledge carrier finder system. This approach has the goal of connecting a knowledge seeker and a knowledge carrier in a logical and simplified way.

Acknowledgements

Application of Fuzzy Cognitive Map in the Performance Management in the tourism industry of Graubunden (Einsatz von Fuzzy Cognitive Maps im strategischen Performance Management der Tourismusbranche Graubunden). This project has been supported, in part, by the Swiss Commission for Technology and Innovation CTI under grant number 14943.1 PFES-ES

References

1. Portmann, E.: The FORA Framework - A Fuzzy Grassroots Ontology for Online Reputation Management. Eds. Andreas Meier, and Edy Portmann. The FORA Framework - A Fuzzy Grassroots Ontology for Online Reputation Management. Springer, Heidelberg: (2013)
2. Berners-Lee; T., Hendler, J., Lassila, O.: The Semantic Web. In: Scientific American Magazine, 284.5, pp. 28–37. (2001)
3. Rappaport, W. J.: What did you mean by that? Misunderstanding, negotiation and syntactic semantics. In: Journal Minds and Machines. Vol. 13, No. 3. pp. 397–427. (2003)
4. Cudre-Mauroux, P.: Emergent Semantics. In: Encyclopedia of Database Systems. Springer, Heidelberg: (2009)
5. Portmann, E., Kaufmann M., Graf C.: A Distributed, Semiotic-Inductive, and Human-Oriented Approach to Web-Scale Knowledge Retrieval. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM 2012). Hawaii: (2012)
6. Ashby, W.R.: An introduction in cybernetics. Chapman & Hall Ltd., London. (1956)
7. Shneiderman, B., Plaisant, C.: Designing the User Interface (4th Ed.). Person/Addison-Wesley, Boston: (2005).
8. Kacprzyk, J.: Computational Intelligence and Soft Computing: Closely Related but Not the Same. In: Accuracy and Fuzziness. A Life in Science and Politics: A Festschrift book to Enric Trillas Ruiz by R. Seising and L. A. Argiles Mndez (Ed.). Springer, Heidelberg: (2015)
9. Jazzar, M., Bin Jantan, A.: Using Fuzzy Cognitive Maps to Reduce False Alerts in SOM-based Intrusion Detection Sensors. In: Proceedings of the 2nd Asia International Conference on Modelling & Simulation. (2008)
10. Aguilar, J.: A survey about fuzzy cognitive maps papers (Invited Papers). In: International Journal Computing Cognition 3, pp.27–33. (2005)
11. Stach, W., Kurgan, L., Pedrycz, W., Reformat, M.: Genetic learning of fuzzy cognitive maps. In: Fuzzy Sets and Systems 153. pp.371–401 (2005)
12. Dickerson, J. A., Kosko, B.: Virtual Worlds as fuzzy cognitive maps, Presence 3 (2) pp.173–189. (1994)
13. Papageorgiou, E. I., Stylios, C. D., Groumpos, P.P., Vrahatis, M. N.: Fuzzy cognitive maps learning using swarm optimization. In: Journal of intelligent information systems. (2005)
14. Papageorgiou, E. I., Stylios, C. D., Groumpos, P.P.: Fuzzy cognitive map learning based on nolinear Hebbian rule. Australian Conference on Artificial Intelligence. pp.256–268. (2003)

15. Papageorgiou, E. I., Stylios, C. D., Groumpos, P.P.: Active Hebbian learning algorithm to train fuzzy cognitive maps. In: *International Journal in Approx. Reasoning*, pp.219–2249. (2004)
16. Papageorgiou, E. I., Stylios, C. D.: Fuzzy cognitive maps. *Handbook of Granular Computing*. John Wiley & Son Ltd, Publication Atrium, Chichester, England (2008).
17. Khan, M. S., Chong, A.: Fuzzy cognitive maps with genetic algorithm. In: *Proceedings of the 1st Indian Conference on Artificial Intelligence*. (2003).
18. Zadeh, L. A.: Outline of a new approach to the analysis of complex systems and decision processes. In: *IEEE Trans. SMC, SMC-3 (1)*, pp. 28–44. (1973)
19. Zadeh, L. A.: Fuzzy sets. In: *Information and Control*, vol. 8, issue 3, pp. 338-353. (1965)
20. Kosko, B.: Fuzzy cognitive maps. In: *Int. J. Man-Machine Studies*, vol. 24, pp. 65-75. (1986)
21. Axelrod, R.: *Structure of Decision: the Cognitive Maps of Political Elites*. Princeton University Press, Princeton. (1976)
22. Kohonen, T., Honkela, T.: Kohonen network. In: *Scholarpedia*, 2(1):7421, (2007)
23. Bibliography of SOM Papers, <http://www.cis.hut.fi/research/som-bibl>. Accessed (10-08-2015)
24. Daisuke, N., O. Matashige. Application of spherical SOM in clustering. In: *Proceedings of Workshop on Self-Organizing Maps, WSOM03, Kitakyushu, Japan*. (2003)
25. Farid, B., Biela, E. P., Jack-Grard P.: Self organizing spherical map architecture for 3d object modeling. In: *Proceedings of Workshop on Self-Organizing Maps, WSOM03, Japan*. (2003).
26. Fritzke, B.: Growing grid - a self organizing network with constant neighborhood range and adaption strength. In: *Neural Processing Letter*, vol. 2, no.5, pp.9–13 (1995)
27. Miikkulainen, R.: Script recognition with hierarchical feature maps. In: *Connection Science*, vol. 2, pp.83–101. (1990)
28. Rauber, A., Dittenbach, M., Merkl, D.: The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. In: *IEEE Transactions on Neural Networks*, vol. 13, n. 6, pp. 1331–1341. (2002)
29. Koikkalainen, P.: Tree structured self-organizing maps. In: *Kohonen Maps*, Oja E., Kaski S. Amsterdam, Netherlands: Elsevier, pp.121–130. (1999)
30. Ultsch, A.: Emergence in Self-Organizing Feature Maps. In: Ritter, H., Haschke, R.: *Proceedings of the 6th International Workshop on Self-Organizing Maps*. (2007)
31. Robinson, I., Webber J., Eifrem E.: *Graph Databases*. OReilly Media Inc, pp. 50–60. (2013)
32. Buerli, M.: *The Current State of Graph Databases*. Departement of Computer Science, Cal Poly, San Luis Obispo. (2012)
33. Shao, B., Wang, H., Xiao Y.: Managing and mining large graphs: systems and implementations. In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD 12*, p. 589–592, ACM, New York. (2012)
34. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: *arXiv 1301.3781v3*, Cornell University Library, New York. (2013)
35. Maas, A., Daly, R., Pham, P., Huang D., Ng, A., Potts C.: Learning Word Vectors for Sentiment Analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol 1, pp. 142–150 (2011)

36. Barua, A., Thomas, S., Hassan, A.E.: What are developers talking about? An analysis of topics and trends in Stack Overflow. In: Empirical Software Engineering , vol 19, issue 3, pp. 619–654. (2014)
37. Vasilescu, B.: StackOverflow and GitHub: Associations between Software Development and Crowdsourced Knowledge. In: Social Computing (SocialCom), 2013 International Conference, pp. 188–195. (2013)
38. Asaduzzaman, M. Mashiyat, A., Roy, C., Schneider, K.: Answering questions about unanswered questions of stack overflow. In: MSR '13: Proceedings of the 10th Working Conference on Mining Software Repositories. (2013)
39. Baroni M., Dinu, G., Kruszewski, G. Don't count, predict! A systematic comparison of context-count vs context-predicting semantic vectors. Association of Computer Linguistics. (2014)
40. OrientDB - OrientDB Multi-Model NoSQL Database, <http://orientdb.com/orientdb/>. Accessed (10-08-2015)
41. Prototype FCM on StackOverflow Computer Science Data, <http://knowtology.com/orientdb/>. Accessed (10-08-2015)
42. Denzler, A.; Wehrle, M.; Meier, A.: A granular knowledge cube. World Academy of Science, Engineering and Technology, International Science Index, Computer and Information Engineering. (2015)