Patent Service Self-Organizing Maps

Aviv Segev Jussi Kantola
Department of Knowledge Service Engineering, KAIST
335 Gwahangno (373-1 Guseong-dong), Yuseong-gu,
Daejeon 305-701, Korea
{aviv, jussi} @kaist.edu

***Abstract*—Patent users such as government, inventors, and manufacturing organizations strive to identify the directions in which the new technology is advancing. The organization of patent knowledge in maps aims at outlining the boundaries of existing knowledge. A model based on knowledge extraction from patents and self-organizing maps for knowledge representation is presented. The model was tested on patents from the United States Patent and Trademark Office. The experiments show promising results in classifying main directions of patent development.**

Index Terms*—Knowledge Engineering, Decision Support, Patent Service, Self-Organizing Maps

I. Introduction

Government services attempt to forecast main research areas that would be beneficial to fund. Similarly, researchers try to map knowledge and identify possible gaps relevant to the advancement of science. The extraction of relevant information from patents allows the analysis of main research areas and the mapping of the current topics of interest. The creation of such a service, which allows analysis of patents over time, will provide decision makers with a top level overview of the direction of new inventions. In addition, the service could support knowledge seekers in identifying worthwhile research tasks. A knowledge map service can enable a researcher to identify the need for specific research directions considered "hot". In addition, research and government institutions providing funding will be able to preplan with a longer horizon and divert research funds to necessary fields. Knowledge maps of patents can assist in the classification of directions of research in the past and in the attempt to predict future discovery directions.

The patent service is unique compared to other knowledge based services because of the requirement to identify whether similar knowledge exists as opposed to the need to locate knowledge. Contemporary knowledge based services are based on using existing information, while the patent support service is required to assist in identifying similar domains and patterns that would result in the rejection of a patent request. Furthermore, patents in different countries are not classified under one classification system.

The premise of the patent system lies in its mutual benefit to both the inventor and the public. In return for full public disclosure, a patent offers certain rights to an inventor for a limited period of time, during which the inventor may exclude all others from making, using, importing, or selling his or her invention. The patent is published and disseminated to the public so that others may study the invention and improve upon it. The constant evolution of science and technology, spurred by the monetary incentive the patent system offers to inventors, strengthens the economy. New inventions lead to new technologies, create new jobs, and improve our quality of life.

The work analyzes patents to create an outline of knowledge. The research aims at building a model that predicts the identification of new critical research areas that can exponentially speed up the overall research in specific fields. The patent project analyzes patents from the United States Patent and Trademark Office. The patent analysis process outline is presented in Figure 1. The first step includes parsing existing patents text. In the analyzed cases the entire patent description was used. Alternative methods include parsing the patents according to dates or according to topics. The model includes three major steps: patent knowledge extraction, knowledge representation using self-organizing maps, and analysis of the evolution. The patent knowledge extraction extracts key features from each patent. The knowledge representation creates an evolving map using the self-organizing map technique to represent the patents research topics. The last step involves the analysis of the knowledge representation map evolution.

The next section describes the related work. Section 3 describes the knowledge extraction model and Section 4 describes the self-organizing maps model. Section 5 describes the results of initial experiments. Section 6 presents a discussion and some concluding remarks.

II. Related Work

Knowledge Representation

It has been proposed to use a multilevel semantic network to represent knowledge within several levels of contexts [19]. The zero level of representation is a semantic network that includes knowledge about basic domain objects and their relations. The first level of representation uses a semantic network to represent contexts and their relationships. The second level presents relationships of metacontexts, the next level describes metametacontexts, and so forth. The top level includes knowledge considered to be true in all contexts. In this work we do not explicitly limit the number of levels in the semantic

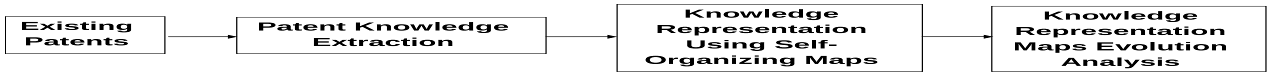

Figure 1 - Patent Process Outline

network. However, due to the limited capabilities of context extraction tools nowadays (see below), we define context as sets of sets of descriptors at zero level only and the mapping between contexts and ontology concepts is represented at level 1. Generally speaking, our model requires n+1 levels of abstraction, where n represents the abstraction levels needed to represent contexts and their relationships.

A previous work on contexts [18] uses metadata for semantic reconciliation. They define the semantic domain of an attribute as the set of attributes used to define its semantics. Work by [7] uses contexts organized as a meet semi-lattice and associated operations like the greatest lower bound for semantic similarity are defined. The context of comparison and the type of abstractions used to relate the two objects form the basis of a semantic taxonomy. They define ontology as the specification of a representational vocabulary for a shared domain of discourse. Both these approaches use ontological concepts for creating contextual descriptions and serve best when creating new ontologies. In this work, we do not focus on ontology generation, which can be performed in any one of various methods, including those mentioned above.

The creation of taxonomies from metadata (in XML/RDF) containing descriptions of learning resources was undertaken in [14]. Following the application of basic text normalization techniques, an index was built, which can be observed as a graph with learning resources as nodes connected by arcs labeled by the index words common to their metadata files. A cluster mining algorithm is applied to this graph and then the controlled vocabulary is selected statistically. A manual effort is necessary to organize the resulting clusters into hierarchies. When dealing with medium-sized corpora (a few hundred thousand words), the terminological network is too vast for manual analysis, and it is necessary to use data analysis tools for processing. Therefore, Assadi [1] employed a clustering tool that utilizes specialized data analysis functions and clustered the terms in a terminological network to reduce its complexity. These clusters are then manually processed by a domain expert to either edit them or reject them.

Several distance metrics were proposed in the literature and can be applied to measure the quality of context extraction. Prior work presented methods based on information retrieval techniques [15] for extracting contextual descriptions from data and evaluating the quality of the process. The Latent Semantic Indexing (LSI) approach presented in the work of [6] and [10] associates word-based vectors to topics in a taxonomy. The underlying idea of LSI is that the aggregate of all the word contexts in which a given word does and does not appear provides a set of mutual constraints that largely determines the similarity of meaning of words and sets of words.

Methods incorporating techniques for analyzing quality of information include Motro and Rakov [13], who proposed a standard for specifying the quality of databases based on the concepts of soundness and completeness. The method allowed the quality of answers to arbitrary queries to be calculated from overall quality specifications of the database. Another approach [11] is based on estimating loss of information in navigating ontological terms. The measures for loss of information were based on metrics such as precision and recall on extensional information. These measures are used to select results having the desired quality of information and we shall use them in our empirical evaluation as well.

In an ongoing work in the European Union called PATexpert [22] several areas of patent services are targeted. The goal of the project is to bring patent services to a new level by applying several new approaches and methods to various areas in patent services. The search method proposed in this article is different from the approach described in PATexpert. In PATexpert the classification process is manual. In our method the classification is a semi-automatic process.

Self-Organizing Maps

The Self-Organizing Map (SOM) is a two-layer unsupervised neural network that maps multidimensional data onto a two dimensional topological grid [9]. The data are grouped according to similarities and patterns found in the dataset, using some form of distance measure, usually the Euclidean distance. The results are displayed as a series of nodes on the map, which can be divided into a number of clusters based upon the distances between the clusters. Since the SOM is unsupervised, no target outcomes are provided, and the SOM is allowed to freely organize itself, based on the patterns identified, making the SOM an ideal tool for exploratory data analysis.

According to Kaski and Kohonen [8], exploratory data analysis methods, such as SOM, are like general-purpose instruments that illustrate essential features of a data set, such as clustering structure and relations between its data items. The SOM perform visual clustering of data [3]. Back et al. [2] provide more information about the methodology of applying self-organizing maps. The most commonly used method for visualizing the final self-organizing map is the unified distance matrix method, or U-matrix [21]. The U-matrix method can be used to discover otherwise invisible relationships in a high-dimensional data space. It also makes it possible to classify data sets into clusters of similar values. Feature planes, representing the values in a single vector column, are used to identify the characteristics of these clusters [3].

III. Patent Knowledge Extraction

Each claim is analyzed separately through patent knowledge extraction. To analyze the claims, a context extraction algorithm and a term frequency/inverse document frequency algorithm can be used. To handle the different vocabularies used by different information sources, a comparison based on context is used in addition to simple string matching.

Context Extraction

We define a context descriptor c_i from domain DOM as an index term used to identify a record of information [12], which in our case is a patent. It can consist of a word, phrase, or alphanumerical term. A weight $w_i \epsilon R$ identifies the importance of descriptor c_i in relation to the patent. For example, we can have a descriptor $c_1 = Length$ and $w_1 = 2$. A descriptor set $\{<c_i,w_i>\}$ is defined by a set of pairs, descriptors and weights. Each descriptor can define a different point of view of the concept. The descriptor set eventually defines all the different perspectives and their relevant weights, which identify the importance of each perspective.

By collecting all the different viewpoints delineated by the different descriptors, we obtain the context. A context $C = \{\{<c_{ij},w_{ij}>\}_i\}_j$ is a set of finite sets of descriptors, where i represents each context descriptor and j represents the index of each set. For example, a context C may be a set of words (hence *DOM* is a set of all possible character combinations) defining a patent and the weights can represent the relevance of a descriptor to the patent. In classic Information Retrieval, $<c_{ij},w_{ij}>$ may represent the fact that the word c_{ij} is repeated w_{ij} times in the patent.

The Patent Knowledge Extraction process uses the World Wide Web as a knowledge base to extract multiple contexts for the textual information. The algorithm input is defined as a set of textual propositions representing the claim information description. The result of the algorithm is a set of contexts - terms that are related to the propositions. The context recognition algorithm was adapted from [17] and consists of the following three steps:

1) Context retrieval: Submit each parsed claim to a Web-based search engine. The contexts are extracted and clustered from the results.

2) Context ranking: Rank the results according to the number of references to the keyword, the number of Web sites that refer to the keyword, and the ranking of the Web sites.

3) Context selection: Assemble the set of contexts for the textual proposition, defined as the outer context.

The algorithm then calculates the sum of the number of Web pages that identify the same descriptor and the sum of number of references to the descriptor in the patent. A high ranking in only one of the weights does not necessarily indicate the importance of the context descriptor. For example, high ranking in only Web references may mean that the descriptor is important since the descriptor widely appears on the Web, but it might not be relevant to the topic of the patent.

The weight of each context can be determined according to the number of retrieved Web references related to the concept or the number of references to the concepts in the patents. Alternatively, the weight can contribute equally to both the number of Web references and the number of patent references to the concept. Another option is setting the weight as the square root of the sum of the number of Web references squared and the number of patent references squared.

Term Frequency / Inverse Document Frequency

The external weight of each context is determined according to the number of retrieved Web references related to the concept and the number of references to the concepts in the patents. In addition, the Term Frequency/Inverse Document Frequency (TF/IDF) method analyzes the patent from an internal point of view, i.e., what concept in the text best describes the patent.

TF/IDF is a common mechanism in IR for generating a robust set of representative keywords from a corpus of documents. The method is applied here to the patent documents. By building an independent corpus for each document, irrelevant terms are more distinct and can be thrown away with a higher confidence. To formally define TF/IDF, we start by defining $freq(t_i,D_i)$ as the number of occurrences of the term t_i within the document D_i. We define the term frequency of each term t_i as:

$$tf(t_i) = \frac{freq(t_i, \mathcal{D}_i)}{|\mathcal{D}_i|}$$

We define D_{patent} to be the corpus of patent documents. The inverse document frequency is calculated as the ratio between the total number of documents and the number of documents that contain the term:

$$idf(t_i) = \log \frac{|\mathcal{D}_{patent}|}{|\{\mathcal{D}_i \,:\, t_i \in \mathcal{D}_i\}|}$$

The TF/IDF weight of a term, annotated as $w(t_i)$, is calculated as:

$$w(t_i) = tf(t_i) \times idf^2(t_i)$$

While the common implementation of TF/IDF gives equal weights to the term frequency and inverse document frequency (i.e., $w = tf \ x \ idf$), we chose to give higher weight to the *idf* value. The reason behind this modification is to normalize the inherent bias of the *tf* measure in short documents [16].

IV. Self-Organizing Maps

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network 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. Self-organizing maps are different from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space.

This makes SOMs useful for visualizing low-dimensional views of high-dimensional data, akin to multidimensional scaling. The model was first described as an artificial neural network by Kohonen [9].

Like most artificial neural networks, SOMs operate in two modes: training and mapping. Training builds the map using input examples. It is a competitive process, also called vector quantization. Mapping automatically classifies a new input vector.

A self-organizing map consists of components called nodes or neurons. A weight vector of the same dimension as the input data vectors and a position in the map space are associated with each node. The usual arrangement of nodes is a regular spacing in a hexagonal or rectangular grid. The self-organizing map describes a mapping from a higher dimensional input space to a lower dimensional map space. The procedure for placing a vector from a data space onto the map is to find the node with the closest weight vector to the vector taken from a data space and to assign the map coordinates of this node to our vector.

While it is typical to consider this type of network structure as related to feedforward networks where the nodes are visualized as being attached, this type of architecture is fundamentally different in arrangement and motivation.

Useful extensions include using torodial grids where opposite edges are connected and using large numbers of nodes. It has been shown that while self-organizing maps with a small number of nodes behave in a way that is similar to K-means, larger self-organizing maps rearrange data in a way that is fundamentally topological in character.

It is also common to use the U-Matrix. The U-Matrix value of a particular node is the average distance between the node and its closest neighbors [5]. In a square grid for instance, we might consider the closest nodes, or six nodes in a hexagonal grid.

Large SOMs display emergent properties. In maps consisting of thousands of nodes, it is possible to perform cluster operations on the map itself [4].

Learning Algorithm

The goal of learning in the self-organizing map is to cause different parts of the network to respond similarly to certain input patterns. This is partly motivated by how visual, auditory, or other sensory information is handled in separate parts of the cerebral cortex in the human brain.

The weights of the neurons are either initialized to small random values or sampled evenly from the subspace spanned by the two largest principal component eigenvectors. With the latter alternative, learning is much faster because the initial weights already give a good approximation of SOM weights [20].

The network must be fed a large number of example vectors that represent, as closely as possible, the kinds of vectors expected during mapping. The examples are usually administered several times as iterations.

The training utilizes competitive learning. When a training example is fed to the network, its Euclidean distance to all weight vectors is computed. The neuron with weight vector most similar to the input is called the best matching unit (BMU). The weights of the BMU and neurons close to it in the SOM lattice are adjusted towards the input vector. The magnitude of the change decreases with time and with distance from the BMU. The update formula for a neuron with weight vector $\mathbf{Wv}(t)$ is

$$\mathbf{Wv}(t + 1) = \mathbf{Wv}(t) + \Theta(v, t)\,\alpha(t)(\mathbf{D}(t) - \mathbf{Wv}(t)),$$

where $\alpha(t)$ is a monotonically decreasing learning coefficient and $\mathbf{D}(t)$ is the input vector. The neighborhood function $\Theta(v, t)$ depends on the lattice distance between the BMU and neuron v. In the simplest form it is one for all neurons close enough to the BMU and zero for others, but a Gaussian function is a common choice, too. Regardless of the functional form, the neighborhood function shrinks with time. At the beginning, when the neighborhood is broad, the self-organizing takes place on a global scale. When the neighborhood has shrunk to just a couple of neurons, the weights are converging to local estimates.

This process is repeated for each input vector for a (usually large) number of cycles λ. The network winds up associating output nodes with groups or patterns in the input data set. If these patterns can be named, the names can be attached to the associated nodes in the trained net.

During mapping, there will be one single winning neuron: the neuron whose weight vector lies closest to the input vector. This can be simply determined by calculating the Euclidean distance between input vector and weight vector.

While this article has emphasized representing input data as vectors, it should be noted that any kind of object which can be represented digitally and which has an appropriate distance measure associated with it and in which the necessary operations for training are possible can be used to construct a self-organizing map. This includes matrices, continuous functions, or even other self-organizing maps.

Algorithm

1. Randomize the map's nodes' weight vectors
2. Select an input vector
3. Traverse each node in the map
    1. Use Euclidean distance formula to find similarity between the input vector and the map's node's weight vector
    2. Track the node that produces the smallest distance (this node is the best matching unit, BMU)
4. Update the nodes in the neighborhood of BMU by pulling them closer to the input vector
    1. $\mathbf{Wv}(t + 1) = \mathbf{Wv}(t) + \Theta(t)\alpha(t)(\mathbf{D}(t) - \mathbf{Wv}(t))$
5. Increment t and repeat from 2 while $t < \lambda$

V. Experiments

The experiments included a set of 81 patents randomly selected from the United States Patent and Trademark Office. Each patent included a vector with 43 top ranking context-extracted values. The patents were processed according to the following steps:

- Extracting the context knowledge of each patent using the World Wide Web as a knowledge base.
- Creating a map of the patents according to knowledge extracted using the Self-Organizing Maps.

The set of experiments included:

- Identifying the main clusters of the patents.
- Analyzing the patent maps according to each context to identify meaningful contexts.

The clustering results of the self-organizing maps are presented in Figure 2. The tool used in the experiments was

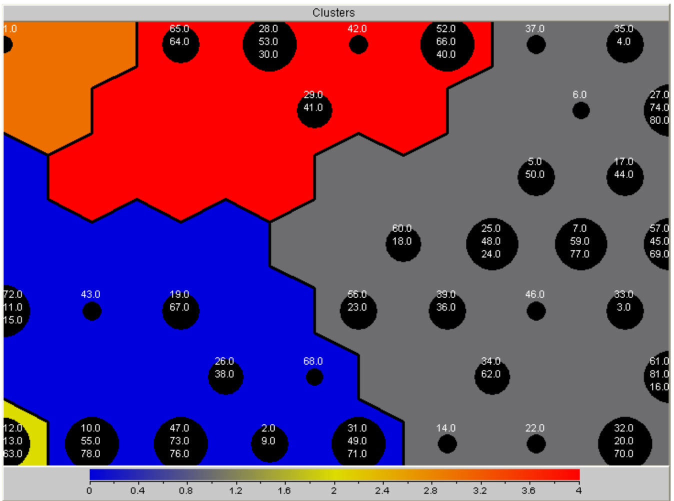

Figure 2 - SOM Patent Clusters

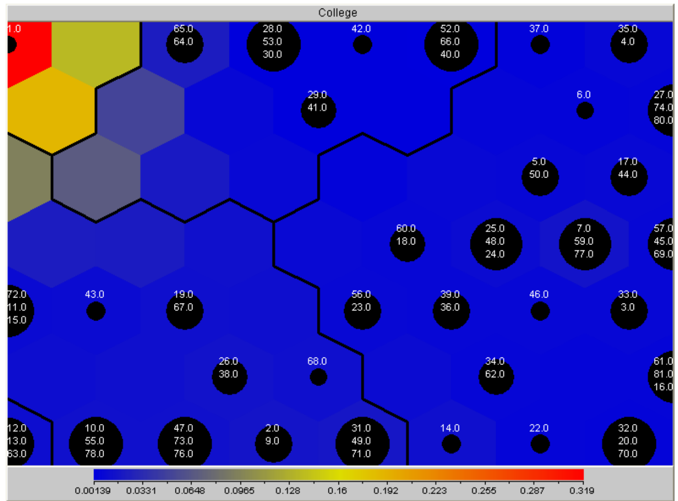

Figure 3 - SOM College Patents

eSom2. The self-organizing maps identified five different clusters. Each cluster is represented by a different color. Each hexagon node in the map represents patents with the closest weight vector to the vector taken from the data space. The circle size represents the number of patents included in that node. The numbers in each circle represent up to three of the patent ids included in the node. It can be seen that one cluster includes only a single patent on the top left. Another cluster includes at least three patents on the bottom left. The other three clusters include multiple patents. Empty nodes indicate that no patents were found for that specific vector distance.

The next analysis tries to identify main context characteristics that classify each patent. In addition, the analysis evaluates how much each context uniquely identifies the patent and the cluster. Each cluster is expected to be identified with a set of contexts.

One example of context classification is displayed in Figure 3, which presents the self-organizing map according to the context *college*. According to the bar on the bottom, the weight relevance of each patent to the specific context can be viewed, when the red marks a high relevance level and the blue a low relevance level. The context *college* uniquely classifies a single patent in the top left corner. Although there are two other nodes in the cluster, none of them include actual patents. The results show that the context college uniquely identifies not only the specific patent but also the cluster. All other patents have no relevance to the patent when analyzed by the point of view of the context of *college*.

A different context classification example of *university* is displayed in Figure 4. The context classifies the two bottom left clusters and in addition the top left cluster. The results show that the university concept of *university* contains the concept of *college*. The cluster on the right also includes patents related to the *university* but with much lower relevance. It can be seen that the concept of *university* is a type of *college*. Another example of a context that identifies a cluster is the image presented in Figure 5. The second cluster from the left identifies many of the patents with the context image. However, the image context does not uniquely classify the cluster since some of the patents identified by values 31, 49,

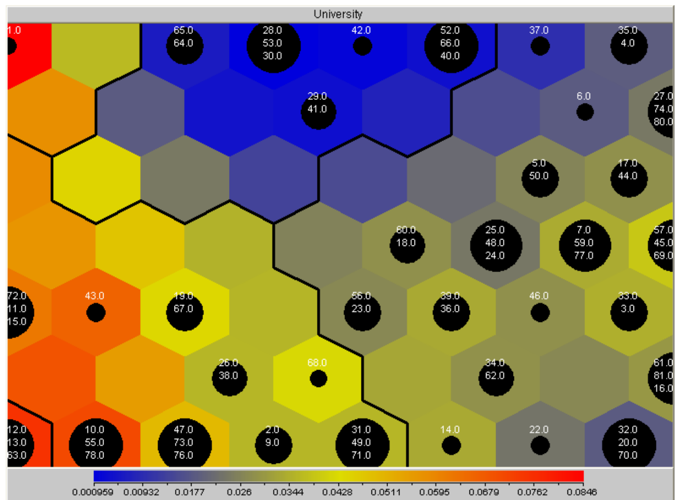

Figure 4 - SOM University Patents

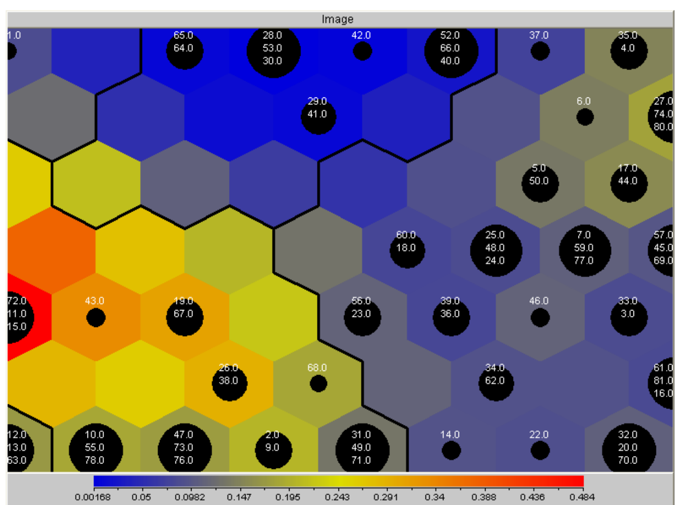

Figure 5 - SOM Image Patents

and 71 are vaguely related to the context. The *image* context is one of the contexts that describe the cluster.

Figure 6 presents the self-organizing map according to the *photo* context. The *photo* context displays a more general term that includes the *image* context map displayed in Figure 5. The *photo* related patents expand past a single cluster. In addition, the *photo* concept includes the concept of *image* as well as more information that is presented in other concepts.

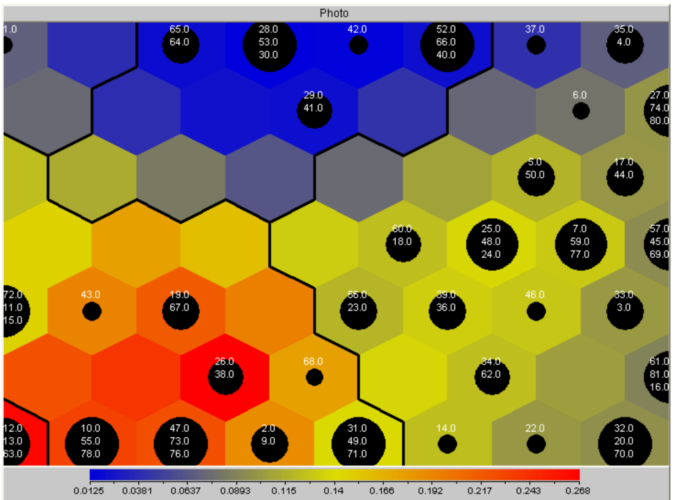

Figure 6 - SOM Photo Patents

The self-organizing map results present the ability to automatically extract relevant contexts and their relevance to classify patents. The results suggest that some contexts can uniquely identify a specific patent and a specific cluster. Other context descriptors extend past a single cluster and suggest relations between patents.

VI. Conclusion and Future Work

The patent service model described in the paper allows a self-organizing map to be created on the boundaries of existing knowledge. The model shows promise in extending the field of patent service. This paper describes a work-in-process, and we are currently working on validating and expanding the data set of the proposed joined approach.

We are in the process of verifying the results predicted by SOM by tagging the patents according to their year and evaluating which contexts have become realities for those patents that are a few years old. We may be able to say something about the predicting power of context SOMs and about the "lead time" from patent context to reality.

In this work, SOM produces results based on existing patents. Therefore a question arises on how we can forecast something that lies ahead in the future, based on the existing dataset embedded in the SOM. When we use context time-series in SOM we may see tendencies that show us a way that leads to "out-of-scope" knowledge or to knowledge that is beyond existing knowledge. If we can get some additional hint or clue on what will be important in the future, we can be proactive in many ways. We believe that the usefulness of the proposed patent context self-organizing maps is in the increased change to see what will be important in the future.

Future work involves evaluating the patent search model against patents over a timeline to evaluate change in knowledge. Another direction is to extend the model to multiple languages.

References

1. Assadi, H.: Construction of a regional ontology from text and its use within a documentary system. In: Proceedings of the International Conference on Formal Ontology and Information Systems (FOIS-98) (1998)
2. Back B., Sere K., Vanharanta H., Managing complexity in large data bases using self-organizing maps. Accounting Management and Information Technologies, 8, 4, pp. 191–210 (1998)
3. Eklund T., Back B., Vanharanta H., Visa A., Using the self-organizing map as a visualization tool in financial benchmarking. Information Visualization, 2, 3, pp. 171–181 (2003)
4. Haykin S., Self-organizing maps, In: Neural networks - A comprehensive foundation (2nd ed.). Prentice-Hall (1999)
5. Heskes T., Energy functions for self-organizing maps, In: Kohonen Maps, E. Oja, S. Kaski (Eds.), Elsevier (1999)
6. Kashyap, V., Dalal, S., Behrens, C.: Professional services automation: A knowledge management approach using LSI and domain specific ontologies. In: Proceedings of the 14th International FLAIRS Conference (Florida AI Research Symposium), Special track on AI and Knowledge Management (2001)
7. Kashyap, V., Sheth, A.: Semantic and schematic similarities between database objects: a context-based approach. VLDB Journal 5, 276–304 (1996)
8. Kaski S., Kohonen T., Exploratory data analysis by the self-organizing map: structures of welfare and poverty in the world. In: Apostolos PN, Refenes YA-M, Moody J., Weigend A. (eds.) Neural Networks in Financial Engineering. Singapore, World Scientific, 498–507 (1996)
9. Kohonen T., Self-Organizing Maps. Springer-Verlag, Leipzig, Germany (2001)
10. Liu, T., Chen, Z., Zhang, B., Ma, W.-Y., Wu, G.: Improving text classification using local latent semantic indexing. In: Perner, P. (ed.) ICDM 2004. LNCS (LNAI), vol. 3275, pp. 162–169. Springer, Heidelberg (2004)
11. Mena, E., Kashyap, V., Illarramendi, A., Sheth, A.P.: Imprecise answers in distributed environments: Estimation of information loss for multi-ontology based query processing. International Journal of Cooperative Information Systems 9(4), 403–425 (2000)
12. Mooers C., Encyclopedia of Library and Information Science. Marcel Dekker, vol. 7, ch. Descriptors, pp. 31–45 (1972)
13. Motro, A., Rakov, I.: Estimating the quality of databases. Lecture Notes in Computer Science (1998)
14. Papatheodorou, C., Vassiliou, A., Simon, B.: Discovery of ontologies for learning resources using word-based clustering. In: Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2002), pp. 1523–1528 (2002)
15. Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworths, London (1979)
16. Robertson S., Understanding inverse document frequency: On theoretical arguments for IDF, Journal of Documentation, vol. 60, no. 5, pp. 503–520 (2004)
17. Segev A., Leshno M., and Zviran M., Context recognition using Internet as a knowledge base, Journal of Intelligent Information Systems, 29(3) (2007)
18. Siegel, M., Madnick, S.E.: A metadata approach to resolving semantic conflicts. In: Proceedings of the 17th International Conference on Very Large Data Bases, pp. 133–145 (1991)
19. Terziyan, V., Puuronen, S.: Reasoning with multilevel contexts in semantic metanetwork. In: Nossun, R., Bonzon, P., Cavalcanti, M. (eds.) Formal Aspects in Context, pp. 107–126. Kluwer Academic Publishers, Dordrecht (2000)
20. Ultsch A., U*-Matrix: a tool to visualize clusters in high dimentional data. University of Marburg, Department of Computer Science, Technical Report Nr. 36:1-12 (2003)
21. Ultsch A., Self organized feature planes for monitoring and knowledge acquisition of a chemical process. The International Conference on Artificial Neural Networks. Springer-Verlag, London, pp. 864–867 (1993)
22. Wanner, L., Baeza-Yatesa, R., Brügmann, S., Codina, J., Diallo, B., Escorsa, E., Giereth, M., Kompatsiaris, Y., Papadopoulos, S., Pianta, E., Piella, G., Puhlmann, I., Raoe, G., Rotard, M., Schoester, P., Serafini, L. and Zervaki, V., Towards content-oriented patent document processing, World Patent Information, 30(1), pp. 21-33 (2008)