Expert Systems with Applications 39 (2012) 13235–13242

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Identification of trends from patents using self-organizing maps

Aviv Segev [a,*], Jussi Kantola [b]

[a] Department of Knowledge Service Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, South Korea
[b] Department of Production, University of Vaasa, P.O. Box 700, Vaasa FI-65101, Finland

ARTICLE INFO

Keywords:
Trend identification
Knowledge representation
Self-organizing maps

ABSTRACT

Patent users such as governments, inventors, and manufacturing organizations strive to identify the directions in which new technology is advancing, and their goal is to outline the boundaries of existing knowledge. The paper analyzes patent knowledge to identify research trends. 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 that the method provides both an overview of the directions of the trends and a drill-down perspective of current trends.

1. 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 a service that 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 can support knowledge seekers in identifying worthwhile research tasks. A knowledge map of patent trends can enable a researcher to identify the need for specific research in a field considered “hot”. In addition, research and government institutions providing funding can 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 directions of discovery.

The patent knowledge trend approach is unique compared to other knowledge based systems because of the requirement to identify whether similar knowledge exists as opposed to the need to locate knowledge. Contemporary knowledge based systems 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, and thus is difficult to identify trends.

The premise of the patent 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 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 identify research trends. The research aims at building a model for the identification of new critical research areas. The identification of new research areas can lead to increased investment of resources by both government and academia and speed up the overall research in specific fields. The patent knowledge trends technique analyzes patents from the United States Patent and Trademark Office. The patent analysis model outline is presented in Fig. 1. The model consists of three major steps: patent knowledge extraction, knowledge representation using self-organizing maps, and knowledge analysis for trend identification. The first step is patent knowledge extraction. The present work used the entire patent description. Alternative methods include parsing the patents according to dates or according to topics. The patent knowledge extraction step extracts key features from each patent. The second step, knowledge representation, creates an evolving map using the self-organizing map technique to represent the patent research topics. This step analyzes the evolution of the knowledge representation using a self-organizing map. The last step, the identification of research trends, analyzes the knowledge extracted in the map and the relevant extracted patent research topics.

The method presents the ability to identify underlying trends that emerge between currently existing clusters and research trends that overshadow the main classification topics currently used. The self-organizing maps allow both an overview of the trend results and a drill-down perspective of current trends.

* Corresponding author.
E-mail addresses: aviv@kaist.edu (A. Segev), jussi.kantola@uwasa.fi (J. Kantola).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.05.078

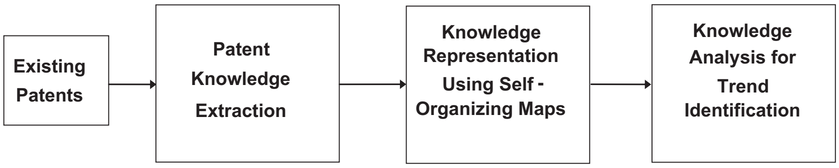
Existing Patents
Patent Knowledge Extraction
Knowledge Representation Using Self - Organizing Maps
Knowledge Analysis for Trend Identification

Fig. 1. Patent analysis model.

The remainder of the paper is organized as follows. The next section describes the related work. Section 3 describes the model for identification of trends from patents. Section 4 describes the results of the experiments. Section 5 presents a discussion and some concluding remarks.

2. Related work

2.1. Knowledge representation

It has been proposed to use a multilevel semantic network to represent knowledge within several levels of contexts (Terziyan & Puuronen, 2000). 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 network. However, due to the limited capabilities of context extraction tools nowadays, we define context as 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 uses metadata for semantic reconciliation (Siegel & Madnick, 1991). They define the semantic domain of an attribute as the set of attributes used to define its semantics. Work by Kashyap and Sheth (1996) 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. Other methods aim at the construction of ontologies from object-oriented database (Zhanga, Maa, & Yanb, 2011) and knowledge retrieval using ontology mining (Tao, Li, & Nayak, 2008). 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 Papatheodorou, Vassiliou, and Simon (2002). 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 (1998) employed a clustering tool that utilizes specialized data analysis functions and clustered the terms in a terminological network to reduce 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 (Rijsbergen, 1979) for extracting contextual descriptions from data and evaluating the quality of the process. The Latent Semantic Indexing (LSI) approach presented in the work of Kashyap, Ramakrishnan, Thomas, and Sheth (2005) and Liu, Chen, Zhang, Ma, and Wu (2004) 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 the method of Motro and Rakov (1998), 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 (Mena, Kashyap, Illarramendi, & Sheth, 2000) 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. Ontology-based inference can be used for causal explanation, (Besnard, Cordier, & Moinard, 2008) which can be implemented in areas such as the biomedical domain (Pathak, Johnson, & Chute, 2009). These measures are used to select results having the desired quality of information, and we will use them in our empirical evaluation as well.

Various aspects of patent knowledge representation have been addressed. An ongoing work in the European Union called PATExpert (Wanner et al., 2008) targets several areas of patent services. Research has been performed on property-function based patent networks using an analysis of patent contents (Yoon & Kim, 2012), identification of patent vacuums (Son, Suh, Jeon, & Park, 2012), and design patent map visualization display (Chen, 2009). However, these approaches are based on similarity of innovation concepts among patents, while the present approach employs semantic similarity. In addition, these approaches require manual categorization into representative groups, while we propose an automatic process for this step.

2.2. 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 (Kohonen, 2001). 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 (1996), 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 performs visual clustering of data (Eklund, Back, Vanharanta, & Visa, 2003). Back, Sere, and Vanharanta (1998) 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 (Ultsch, 1993). The U-matrix method can be used to discover otherwise invisible relationships in a high-dimensional data space. The U-matrix method 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 (Eklund et al., 2003). Initial work suggested that SOM can be used for patent classification (Segev & Kantola, 2010).

Other directions of multidimensional data analysis include the identification of uncertain dependencies between structural action and response processes using neural networks (Freitag, Graf, & Kaliske, 2011) and the classification of decisions using probabilistic neural network (Ahmadlou & Adeli, 2010). Current work uses SOM to analyze patent trends.

3. Patent trend analysis model

The patent trend analysis model has three steps. The first step, described in Section 3.1, is to extract the context representing the knowledge of each patent. Section 3.2 presents the second step, knowledge representation using the self-organizing maps algorithm and displays the map learning algorithm. Finally, Section 3.3 presents the trend analysis algorithm.

3.1. Patent knowledge extraction

Each patent claim is analyzed separately through patent knowledge extraction. To analyze the claims, a Web context extraction algorithm or 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.

We define a context descriptor c_i from domain $\mathcal{DOM}$ as an index term used to identify a record of information, (Mooers, 1972) which in our case is a patent. It can consist of a word, phrase, or alphanumerical term. A weight $w_i \in \Re$ identifies the importance of descriptor c_i in relation to the patent. For example, we can have a descriptor $c_1 = \textit{fingerprint}$ and $w_1 = 17$. A descriptor set $\{\langle c_i, w_i \rangle\}_i$ 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 view points delineated by the different descriptors, we obtain the context. A context $C = \{\{\langle c_{ij}, w_{ij} \rangle\}_i\}_j$ 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 $\mathcal{C}$ may be a set of words (hence $\mathcal{DOM}$ 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, $\langle c_{ij}, w_{ij} \rangle$may represent the fact that the word c_{ij} is repeated w_{ij} times in the patent.

The context extraction can be based on an external knowledge source such as the Web or an internal source such as the term frequency in the patent text. Both possible methods are described next.

3.1.1. Web context extraction

The Patent Knowledge Extraction process can use the Web as a knowledge base for extracting the patent as multiple descriptors 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 the patent – terms that are related to the propositions. The context recognition algorithm was adapted from Segev, Leshno, and Zviran (2007) and consists of the following three steps:

(i) Context retrieval: submit each parsed claim to a Web-based search engine. The descriptors are extracted and clustered from the results.
(ii) Context ranking: rank the results according to the number of references to the descriptor, the number of Web sites that refer to the descriptor, and the ranking of the Web sites.
(iii) Context selection: assemble the set of descriptors 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 descriptor or the number of references to the descriptor in the patents. Alternatively, the weight can contribute equally to both the number of Web references and the number of patent references to the descriptor. 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.

The external weight of each context is determined according to the number of retrieved Web references related to the descriptors and the number of references to the descriptors in the patents. Alternatively, the Term Frequency/Inverse Document Frequency (TF/IDF) method can be used to analyze the patent from an internal point of view, i.e., what context in the text best describes the patent.

3.1.2. Term frequency/inverse document frequency

TF/IDF is a common mechanism in information retrieval (IR) for generating a robust set of representative keywords from a corpus of documents, although other methods can be used for classifying text streams by keywords (Yang, Zhang, & Li, 2011). The TF/IDF 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:

$$tf(t_i) = \frac{freq(t_i, D_i)}{|D_i|} \tag{1}$$

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:

$$idf(t_i) = \log \frac{|D_{patent}|}{|\{D_i : t_i \in D_i\}|} \tag{2}$$

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

$$w(t_i) = tf(t_i) \times idf^2(t_i) \tag{3}$$

While the common implementation of TF/IDF gives equal weights to the term frequency and inverse document frequency (*i.e.*, $w = tf \times idf$), 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 (Robertson, 2004).

3.2. Knowledge representation using 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, similar to multidimensional scaling. The model was first described as an artificial neural network by Kaski and Kohonen (1996).

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 SOM consists of components called nodes or neurons. Each node is associated with a weight vector of the same dimension as the input data vectors and a position in the map space. 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, (MacQueen, 1967) larger self-organizing maps rearrange data in a way that is fundamentally topological in character.

The U-matrix (Unified Distance Matrix) is used to visualize the data in a high dimensional space on a 2-D image. The distance between the neighboring neurons gives an approximation of the distance between different parts of the underlying data. When such distances are depicted by the color scale image, the high value colors depict the closely spaced nodes and low value colors indicate the more distant nodes. Thus, groups of high value colors can be considered as clusters, and the low value parts as the boundary regions (Heskes, 1999).

Large SOMs display emergent properties. In maps consisting of thousands of nodes, it is possible to perform cluster operations on the map itself (Haykin, 1999).

Our implementation includes a vector from each patent representing all of the relevant descriptors for the patent. The vector represents all of the possible descriptors from all of the patents and the weight describes the relevance of the descriptor in the specific patent. The SOM output displays the map of all topics extracted in Section 3.1 organized according to clusters of topics appearing in multiple patents with high relevance weight value.

3.2.1. Map 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 (Ultsch, 2003).

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 $Wv(t)$ is

$$Wv(t+1) = Wv(t) + \Theta(v,t)\alpha(t)(D(t) - Wv(t)), \tag{4}$$

where $\alpha(t)$ is a monotonically decreasing learning coefficient and $D(t)$ 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 can also be used. 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 we emphasized representing input data as vectors, it should be noted that any kind of object which can be represented digitally, 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

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

3.3. Knowledge analysis for trend identification

Knowledge analysis for trend identification is performed first on the U-matrix to identify existing trends and then on each context descriptor to identify new trends which expand between existing clusters. The U-matrix value of a particular node is the average distance between the node and its closest neighbors (Heskes, 1999). In a square grid for instance, we might consider the closest 4 or 8 nodes (the Von Neumann neighborhood and Moore neighborhood respectively) surrounding a central cell on a two-dimensional square lattice, or six nodes in a hexagonal grid which is used in our case.

`Existing trend` is identified as a context, C_j, composed of a set of descriptors, c_{ij}, that represent a set of adjacent nodes, n_{ij}, belonging to a single cluster, CL_j, identified by the self-organizing map algorithm.

$$T_{exist} = \{\langle c_{ij}, n_{ij}\rangle | c_{ij} \in n_{ij}, \quad c_{ij} \in C_j, \quad n_{ij} \in CL_j, n_{ij}\ adjacent\} \tag{5}$$

`New trend` is identified as a context that expands on a series of adjacent nodes that expands between clusters.

$$T_{new} = \{\langle c_{ij}, n_{ij}\rangle | c_{ij} \in n_{ij}, \quad c_{ij} \in C_j, \quad n_{ij} \in CL_j, \quad n_{ik} \in CL_k, \\ k \neq j, n_{ij}, n_{ik}\ adjacent\} \tag{6}$$

The trend classification is not necessarily noticeable when viewing all the data separately. To identify a trend that extends past an individual cluster, the analysis should be performed on multiple levels, thus allowing a ''zoom out'' option on the data classification. The ''zoom out'' option can be performed by mapping the data to an ontology, a directed graph with nodes representing concepts and edges representing relationships (Bunge, 1979), such as the US Patent classification. The ontology mapping will allow us to analyze the trend at a low, medium, or high level of classification of the data.

4. Experiments

4.1. Data and metrics

The experiments were performed on a set of 447 patents from the United States Patent and Trademark Office. The patents were selected from 17 sets of topics defined by the US Patent Office. Each patent topic had 15 to 49 patents in the set. For each of the top ranking contexts the values were extracted. The number of the top ranking descriptor values extracted varied from 1 to 7. The patents were processed according to the following steps:

- Extracting the context knowledge of each patent using the TF/IDF method.
- Creating a map of the patents according to knowledge extracted using the self-organizing maps.

The set of experiments included:

- Identification of the main clusters of the patents.
- Analysis of the patent maps according to each context to identify meaningful contexts.
- Analysis of the clustering results compared to other clustering methods.
- Analysis of the clustering compared to the ontology hierarchical classification level.

4.2. Experiment results

Fig. 2 presents the overview of the self-organizing map based on the top 7 context descriptors of each of the patents. The clustering results of the self-organizing maps are presented in Fig. 2 (left). The tool used in the experiments was eSom2. The self-organizing map identified six 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. Empty nodes indicate that no patents were found for that specific vector distance. The clustering identifies one big cluster which dominates most of the patents.

The U-matrix in Fig. 2 (right) displays high values representing related patents in a few areas. One example is the top center of the map, which represents an area of high values of related patents that extends between different clusters. It can also be observed from the U-matrix that many of the patents are concentrated in one part of the big cluster. In other words, many of the patents seem to be related, although according to the US Patent classification there are 17 different topic classifications. The following analysis supplies some insights into these findings.

The next experiment performs a drill-down analysis and tries to identify main context characteristics that classify a cluster. In addition, the analysis evaluates the extent to which each context uniquely identifies the patents and the cluster. An ideal result would be expected to identify each cluster with a set of contexts.

One example of cluster trend classification is displayed in Fig. 3 (top), which presents the feature planes of the self-organizing map according to the context of *boats* and *crankshaft*. 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 results show that the context *boats* and *crankshaft* uniquely identifies not only the specific patents but also the cluster.

A different context classification example, of *encrypting* and *fingerprint*, which expands over three clusters, is displayed in Fig. 3 (middle). The two context descriptors overlap one another and can clearly be labeled as one trend. This example raises the question of why the well defined topic does not receive its own cluster and resides exactly on the overlap of three clusters. The simple solution is that the other descriptors of these patents receive higher values which are distant from one another and therefore form different clusters. However, the overlap of multiple context descriptors can also emphasize that a new trend is forming based on the current intersection of multiple clusters. These types of

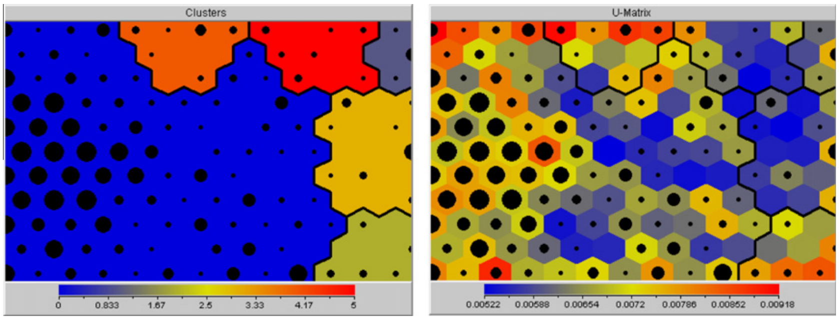

Fig. 2. Clusters and U-matrix.

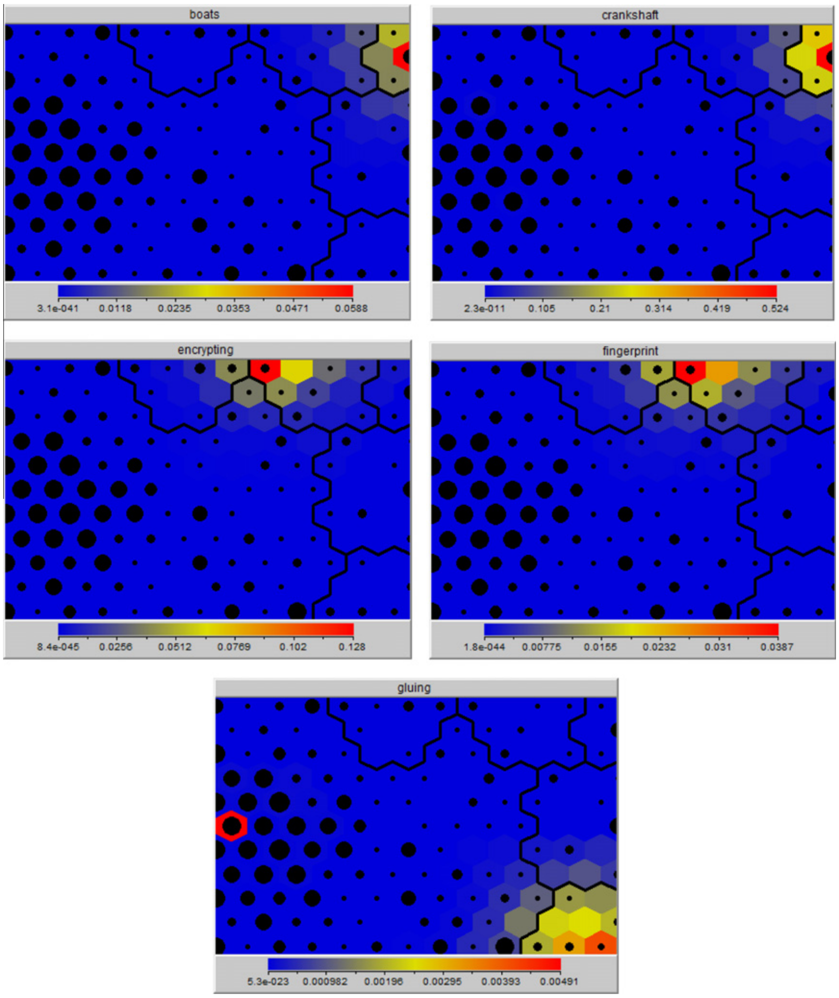

Fig. 3. Patent trend examples.

overlaps can represent new possible trends that can develop over time.

Fig. 3 (bottom) displays an example of a context definition which expands over isolated areas in the self-organizing map. The bottom right identifies one cluster which can be characterized with the context of *gluing*. In addition, at the other side of the map appears a high value representing patents which deal with the same context but are not related to the topic of the cluster. Such examples can be explained by the semantic meaning of *gluing* which can have multiple meanings in different research areas, such as biology or hardware development, based on the material used or on the act of joining.

To analyze the performance of the self-organizing maps, the clustering of two other methods was compared: K-Means and DBSCAN. K-Means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (MacQueen, 1967). DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm which finds a number of clusters starting from the estimated density distribution of corresponding nodes (Ester, Kriegel, Sander, & Xu, 1996).

Fig. 4 displays the F-Measure and Accuracy results for all three clustering methods compared to the US Patent Office human classification of the patents. The F-Measure combines the recall and precision of the results at equal weights. The F-Measure is considered a better evaluation for clustering than analyzing recall since minimizing the number of clusters would achieve a high recall at the cost of low precision. The *X*-axis displays the number of context descriptors extracted from each patent. The *Y*-axis displays the results of each of the methods. The results show that both methods, SOM and K-Means, achieve similar values in the F-Measure evaluation. However, the DBSCAN achieves very low results due to its identification of many of the patent samples as noise.

The Accuracy analysis displays that the SOM method dominates all other methods, followed by the K-Means, and then the DBSCAN. Both the F-Measure and the Accuracy results show that the number of context descriptors extracted have limited effects on the results of the different methods. In other words, increase of the number of context descriptors extracted from each patent does not result in an increase in the F-Measure or Accuracy.

The F-Measure and Accuracy of all three methods yielded relatively low results compared to the human classification clusters defined by the US Patent Office. To analyze the reason for the low results, we evaluated the topic label classification according to the definitions of the US Patent Office. The entire classification of the US Patent Office can be viewed as an ontology. If we analyze the same patents according to the higher level ontology classification, we receive only 13 clusters instead of 17 in the original classification. We defined the original clusters as concepts of the low-level ontology and the new concepts as concepts of the mid-level ontology. We repeated the classification by manually selecting similar labels topics, such as: *electrical computers and digital processing systems, data processing, registers, error detection/correction and fault detection/recovery*, and classified them into one cluster labeled

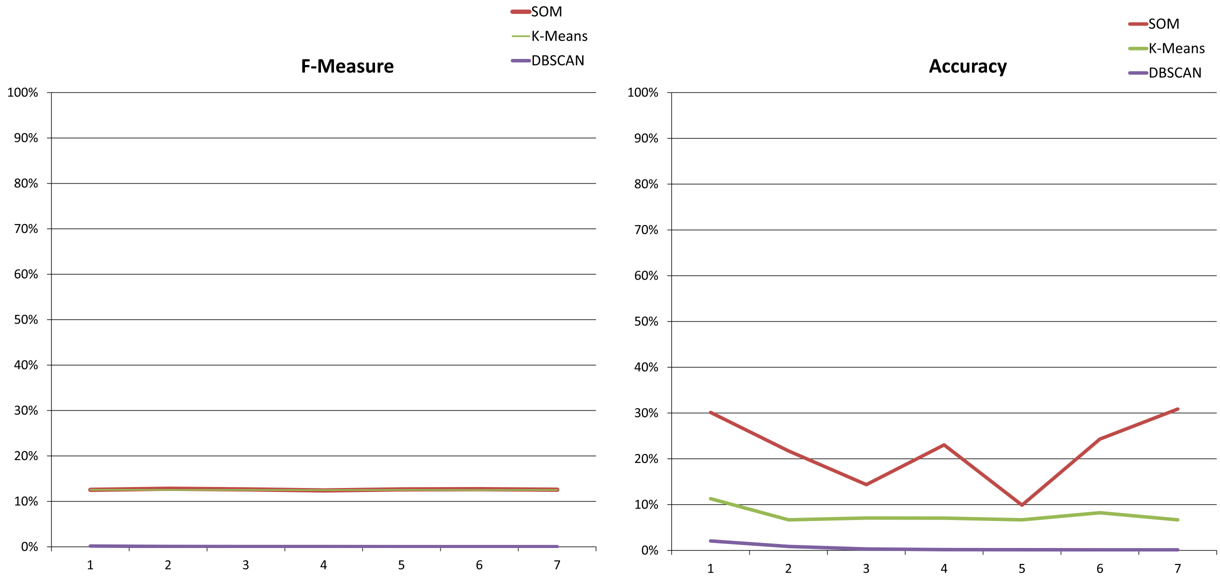

Fig. 4. Methods performance.

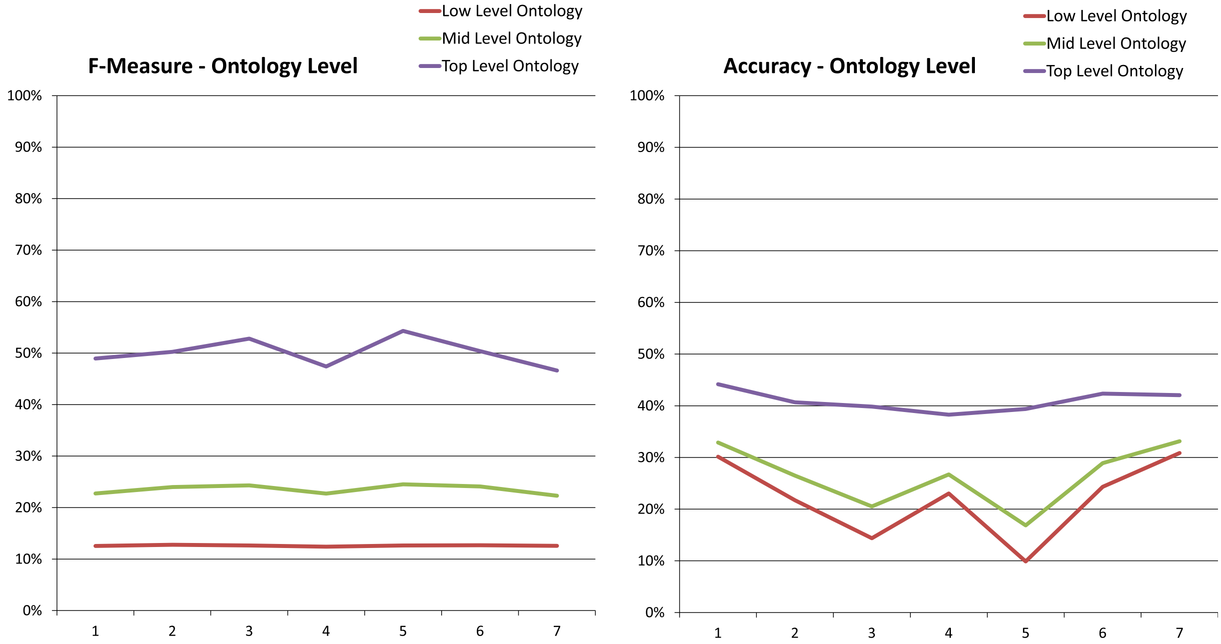

Fig. 5. SOM ontology level performance.

computer architecture. This step resulted in a classification of only 9 concepts which we defined as the top-level ontology.

Fig. 5 displays the F-Measure and Accuracy of the low-level, mid-level, and top-level ontology with 1 to 7 context descriptors extracted from each patent according to the self-organizing map method. The results show a constant increase in the performance of both the F-Measure and Accuracy as the level of ontology is higher. It is interesting that the results imply that the hierarchical level of clusters defined by the US Patent Office might be misclassified and that there is a more general cluster presiding over many of the clusters. This cluster presides over many of the patents classified. If we compare the results to the U-matrix in Fig. 2 (right), we can see that the results suggest that there is one trend which encompasses most of the patents analyzed and forms one big cluster, with many of the patents falling within this "hot" area.

The results of the trend identification method show that the method can automatically extract relevant contexts 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.

5. Conclusion and future work

The patent trend analysis model described in the paper maps existing knowledge in order to identify main research trends. The model shows promise in extending the field of identification of research trends using patents. This paper describes a method based on knowledge extraction from patents and on self-organizing maps for knowledge representation. Relevant information is extracted from patents and the main current topics of interest are mapped according to research areas.

The results of the experiments display that the patent trend analysis model based on the SOM clustering method achieves higher results in accuracy than do the K-Means and DBSCAN clustering methods. In addition, the patent trend analysis method displays the ability to present underlying trends that emerge between currently existing clusters and research trends that overshadow the main classification topics currently used. The patent trend analysis model uses self-organizing maps to allow both an overview of the trend results and a drill-down perspective of current trends.

Future work includes verifying the results predicted by the patent trend analysis method by tagging the patents according to their year and evaluating which trends have become realities for those patents that are a few years old. This work will allow the analysis of the predicting power of context patent trend analysis and the evaluation of the "lead time" from patent context to reality.

References

Ahmadlou, M., & Adeli, H. (2010). Enhanced probabilistic neural network with local decision circles: A robust classifier. *Integrated Computer-Aided Engineering, 17*(3), 197–210.

Assadi, H. (1998). 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)*.

Back, B., Sere, K., & Vanharanta, H. (1998). Managing complexity in large data bases using self-organizing maps. *Accounting Management and Information Technologies, 8*(4), 191–210.

Besnard, P., Cordier, M. O., & Moinard, Y. (2008). Ontology-based inference for causal explanation. *Integrated Computer-Aided Engineering, 15*(4), 351–367.

Bunge, M. (1979). Treatise on basic philosophy. In D. Reidel (Ed.). *Ontology II: A world of systems* (Vol. 4). New York, NY: Publishing Co., Inc.

Chen, R. (2009). Design patent map visualization display. *Expert Systems with Applications, 36*(10), 12362–12374.

Eklund, T., Back, B., Vanharanta, H., & Visa, A. (2003). Using the self-organizing map as a visualization tool in financial benchmarking. *Information Visualization, 2*(3), 171–181.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining (KDD-96)* (pp. 226–231).

Freitag, S., Graf, W., & Kaliske, M. (2011). Recurrent neural networks for fuzzy data. *Integrated Computer-Aided Engineering, 18*(3).

Haykin, S. (1999). *Neural networks – A comprehensive foundation* (2nd ed.). Prentice-Hall. ch. Self-organizing maps, pp. 443–483.

Heskes, T. (1999). *Kohonen maps*. Elsevier. ch. Energy functions for self-organizing maps, pp. 303–316.

Kashyap, V., Ramakrishnan, C., Thomas, C., & Sheth, A. (2005). TaxaMiner: An experimentation framework for automated taxonomy bootstrapping. *International Journal of Web and Grid Services, 1*(2), 240–266 [Special issue on semantic web and mining reasoning].

Kashyap, V., & Sheth, A. (1996). Semantic and schematic similarities between database objects: A context-based approach. *VLDB Journal, 5*, 276–304.

Kaski, S., & Kohonen, T. (1996). *Neural networks in financial engineering*. World Scientific. ch. Exploratory data analysis by the self-organizing map: Structures of welfare and poverty in the world, pp. 498–507.

Kohonen, T. (2001). *Self-organizing maps*. Springer-Verlag.

Liu, T., Chen, Z., Zhang, B., Ma, W. Y., & Wu, G. (2004). Improving text classification using local latent semantic indexing. In *ICDM* (pp. 162–169).

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley symposium on mathematical statistics and probability* (pp. 281–297).

Mena, E., Kashyap, V., Illarramendi, A., & Sheth, A. P. (2000). 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.

Mooers, C. (1972). *Encyclopedia of Library and Information Science* (Vol. 7). Marcel Dekker. ch. Descriptors, pp. 31–45.

Motro, A., & Rakov, I. (1998). Estimating the quality of databases. *Lecture Notes in Computer Science*.

Papatheodorou, C., Vassiliou, A., & Simon, B. (2002). 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).

Pathak, J., Johnson, T. M., & Chute, C. G. (2009). Modular ontology techniques and their applications in the biomedical domain. *Integrated Computer-Aided Engineering, 16*(3), 225–242.

Rijsbergen, C. J. V. (1979). *Information retrieval* (2nd ed.). London: Butterworth.

Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation, 60*(5), 503–520.

Segev, A., & Kantola, J. (2010). Patent search decision support service. In *Proceedings of international conference on information technology: New generations (ITNG 2010)* (pp. 568–573).

Segev, A., Leshno, M., & Zviran, M. (2007). Internet as a knowledge base for medical diagnostic assistance. *Expert Systems with Applications, 33*(1), 251–255.

Siegel, M., & Madnick, S. E. (1991). A metadata approach to resolving semantic conflicts. In *Proceedings of the 17th international conference on very large data bases* (pp. 133–145).

Son, C., Suh, Y., Jeon, J., & Park, Y. (2012). Development of a GTM-based patent map for identifying patent vacuums. *Expert Systems with Applications, 39*(3), 2489–2500.

Tao, X., Li, Y., & Nayak, R. (2008). Knowledge retrieval model using ontology mining and user profiling. *Integrated Computer-Aided Engineering, 15*(4), 313–329.

Terziyan, V., & Puuronen, S. (2000). Reasoning with multilevel contexts in semantic metanetwork. In R. N. P. Bonzon & M. Cavalcanti (Eds.), *Formal aspects in context* (pp. 107–126). Kluwer Academic Publishers.

Ultsch, A. (1993). Self organized feature planes for monitoring and knowledge acquisition of a chemical process. In *Proceedings of the International Conference on Artificial Neural Networks* (pp. 864–867). Springer-Verlag.

Ultsch, A. (2003). U*-Matrix: A tool to visualize clusters in high dimentional data. University of Marburg, Department of computer science. *Technical report* (Vol. 36, pp. 1–12).

Wanner, L., Baeza-Yatesa, R., Brügmann, S., Codina, J., Diallo, B., Escorsa, E., et al. (2008). Towards content-oriented patent document processing. *World Patent Information, 30*(1), 21–33.

Yang, B., Zhang, Y., & Li, X. (2011). Classifying text streams by keywords using classifier ensemble. *Data and Knowledge Engineering, 70*(9), 775–793.

Yoon, J., & Kim, K. (2012). An analysis of propertyfunction based patent networks for strategic R& D planning in fast-moving industries: The case of silicon-based thin film solar cells. *Expert Systems with Applications, 39*(9), 7709–7717.

Zhanga, F., Maa, Z. M., & Yanb, L. (2011). Construction of ontologies from object-oriented database models. *Integrated Computer-Aided Engineering, 18*(4), 327–347.