SUPPORTING TOPIC MAP CREATION USING DATA MINING TECHNIQUES

Witold Abramowicz, Tomasz Kaczmarek, Marek Kowalkiewicz
Department of Management Information Systems
The Poznan University of Economics,
Poznan, Poland

ABSTRACT

There is an increasing interest in automating creation of semantic structures, especially topic maps, by taking advantage of existing, structured information resources. This article gives a preview of the most popular method – based on RDF triples, and suggests a way to automate topic map creation from unstructured information sources. The method can be applied in information systems development domain when analysing vast unstructured data repositories in preparation for system design, or when migrating large amounts of unstructured data from legacy systems. There are two innovative methods presented in the paper – Term Crawling (TC) and Clustering Hierarchy Projection (CHP), which are applied to build a topic map based on free text documents from local repositories and those downloaded from the Internet. The methods originate from data mining techniques for knowledge discovery. A sample tool, which uses described techniques, has been implemented. The preliminary results that have been achieved on the test collection are presented in concluding sections of the article.

BACKGROUND

In today's enterprises, documentation is spread throughout the whole organisation. The locations may include corporate portals, document management systems, users' private folders, web servers and many others. In order to provide employees with access to all relevant documents one should consider using a common data structure which is able to identify location of any (or almost any) document in the enterprise. One such structure, discussed in this paper, can be topic maps. However, introducing a new data structure imposes a requirement to enter data about all the documents which should be made available, and fill in all the necessary attributes. Due to heterogeneity of data sources, automated techniques are still more a concept than reality. On the other hand, just supporting of implementation may be helpful. When using topic maps as a data structure describing the whole document repository, one may apply concepts from the field of information retrieval. After providing background information, we further explore selected techniques and suggest an extension of some.

TOPIC MAPS

Topic maps as a structuring mechanism for repositories are becoming more and more popular. The phenomenon of topic maps can be observed during many conferences, such as XML Europe, in newsgroup discussions (the Oasis society), and in practice – in many applications throughout the world (Rittershofer 2002; Andersen 2003). The basic idea of topic maps has been described in "The TAO of Topic Maps" (Pepper 2000), a more detailed (and standardized) view has been presented in ISO 13250 (ISO 2000) specification, and one of the first attempts to sketch the idea of using topic maps for structuring content has been made in "Topic Maps for repositories" (Ahmed 2000). We focus in this article on the topic map technology, because we believe it is a potentially interesting data structure for information systems development.

Topic maps as semantic structures can be used in information systems development in several areas. They may be used as a tool supporting information systems design. The semantics carried by a topic map may be used to understand the domain and analyse the requirements for the design. One way of using topic maps would be to identify subjects and patterns that may exist in unstructured data analysed for system modelling. The tools, supporting topic map visualization, help to analyse domain and design database. The other potential application area is migration from unstructured data or text repository to the structured one. Automatically created topic maps provide insight into unknown data structure thereby imposing partial structure constraints. The latter application is further explored throughout this article.

Topic maps enable modelling and representation of knowledge in an interchangeable form, and at the same time they make up a uniform framework for knowledge and information resources management (Pepper 2002a,b). Topic maps constitute a model that is applicable by a wide range of industries. They are helpful in organization and navigation of continuously growing information pools. Current Topic Maps architecture bases on the following elements (after Steve Pepper, we call it the **TAO** of Topic Maps):

Topics – the term topic refers to an element in the topic map that represents the subject being referred to. Topics can be categorized. They can have zero or more topic types and can also have names. The standard names for topics are: base name, display name and sort name. Each topic can have facets – attributes for storing additional information, for example topic profiles. *Topic types* determine structure hierarchy, allowing, among other things, construction of either top-level ontologies, domain specific ontologies, or application specific ontologies based on the ISO 13250 TM standard. *Topic names* – among wide range of names we use in day-to-day activity, the ISO standard describes three of them as follows: Base Name – characterises the topic for internal purposes; it is a required name, Display Name is used for external representation purposes and is optional, Sort Name – allows constructing indexes of topics, useful for sorting purposes and facilitate searching.

Associations – a topic association is a link element, showing relationships between two topics. Association can have types (for example *influenced by*, *required by*, *written in* etc.) and roles (for example *influencer*, *influenced*, *prerequisite*, *result*, *document*, *language*). *Association types* – similarly to topics which are grouped by other topics called topic type, associations are grouped by associations types also represented as topics in the Topic Map. Even with large and complicated structures, navigation, searching, and clarity of user interface functionality are preserved by the possibility of creating custom types of association. One may say that topic types and association types play the same role in the Topic Map. *Association roles* – each topic that is involved in an association is characterised by association role i.e. *influencer* and *influenced*.

Occurrences – occurrences link topics to one or more relevant information resources. An occurrence can be anything, most often it is a URI (Universal Resource Identifier), or document (article, picture, video etc.). Occurrences can have roles and role types (web based training, computer based training, MS Word document, flash animation, knowledge base etc.).
Additionally, the ISO specification of Topic Maps defines the following:

Identity – during the mapping or merging of large scale maps it may occur, that two or more topics with different base names are describing the same concept or fact. Making topic maps structure portable we have to take into consideration such situations. There are properties called public subjects (identity) that describe theme of the topic in a standard way. In case of conflict situation topics with the same identity are merged.

Scope – The extent of the validity of a topic characteristic assignment: the context in which a name or an occurrence is assigned to a given topic, and the context in which topics are related through associations.

Facets – Facets basically provide a mechanism for assigning property-value pairs to information resources. A facet is simply a property; its values are called facet values. Facets are typically used for supplying the kind of metadata that might otherwise have been provided by SGML or XML attributes, or by a document management system. This could include properties such as “language”, “security”, “applicability”, “user level”, “online/offline”, etc. Once such properties have been assigned, they can be used to create query filters producing restricted subsets of resources, for example those whose language is “Italian” and user level is “secondary school student”.

The Topic Maps standard is constantly being developed. They are used for knowledge structuring, for example in digital encyclopaedias, where entries can be linked very flexibly. Navigation in such structure is much easier than navigation in traditional tree-based catalogues. Topic Maps technology is being introduced in web search engines (for example http://www.webbrain.com/) enhancing their efficiency (Pepper 2002a,b) .

At the beginning, the idea, as a descendant of the DocBook (Walsh and Muellner 1999) standard, was to create topic maps from scratch. Therefore, the process of creating topic maps would be to identify resources, try to describe them as it was needed, and link subjects with the other ones (a good introduction to that can be found in the mentioned work of Steve Pepper). One of challenging tasks would be to identify all required occurrences of each topic, and then to associate them with this topic. Recently, the topic maps community has focused on automated topic map creation from already existing resources. The automated process is more "occurrence driven", which means that building mechanisms first analyse the source base (for example a set of HTML files), and then create a topic map. In this approach, occurrences finally play a smaller role – their content is treated as a base for creating topics and associations in the map, and afterwards source occurrences can even be omitted. As it is explained later in the text, in order to create such a map, one needs to have access to some structured sources, or mechanisms for Natural Language Processing (NLP). The aim of this article is to suggest a method for automated topic map creation for unstructured sources with no need to use NLP - this makes the process more flexible (e.g. mostly independent from language of source base).

DATA MINING

(Berry and Linoff 1997) define data mining as *the process of exploration and analysis, by automatic or semiautomatic means, of large quantities of data in order to discover meaningful patterns and rules*. They support this definition in their later publications, such as (Berry and Linoff 2000). According to this, data mining is mainly used for the following six activities:

Classification – assigning analysed objects to predefined classes based on its features. This is mainly used for classifying tuples in relational databases. One has to prepare well defined classes, and afterwards, every attribute of a tuple is tested against the definition. Definitions of classes are made based on an analysis of a training set consisting of preclassified examples. Examples of classification include assigning categories to filtered text documents, credit scoring, assigning customers to predefined segments.

Estimation – as opposed to classification, which results in discrete outcomes, estimation creates continuously valued outcomes. Analysing input data, one comes up with a value for output data, being a continuous variable. Estimation may also be used for classification purposes, when one decides, that output values are compared against a classifying threshold. For examples bank scoring rules may state, that a bank customer will get an equity loan if his score – as derived by data mining rules – will be above 70 points. Other examples of estimation include: estimating the number of children in a family, estimating the value of a piece of real estate.

Prediction – being either classification or estimation, but with one specific attribute – there is no way of checking, whether outcome is true or false at the time of deriving the outcome. One has to wait, because at the derivation time the information needed to prove or falsify the rule is unavailable. Examples include predicting most profitable customers within the next quarter, predicting which customers will order broadband Internet.

Affinity grouping and association rules – its task is to determine facts that occur together. The classic example is one of determining which products are bought together by supermarket customers. Affinity grouping may be used to arrange items on shelves in a store or in a catalogue.

Clustering – is used to segment a group of objects into a number of smaller subgroups (clusters). Clustering does not rely on predefined classes (as opposed to classification), and there are no examples. Objects are grouped together based on their self-similarity. Cluster descriptions may be proved by data miner. Clustering is often used as a first step, before applying some other form of data mining.

The list of six methods of data mining is not a complete list. One may suggest some other forms, which will be used, either separately or in cooperation with other methods, to identify rules in underlying data repository. In further parts of this document authors propose an original method of data mining (or, more specifically, text mining) – term crawling.

AUTOMATED TOPIC MAP CREATION

The issue of automated topic map creation has been widely discussed in many publications in 2002, two years after publishing an ISO standard document for the topic maps. There is currently a number of running projects focusing on creating topic maps from existing data sources, one such example is the Ontopia's MapMaker toolkit. The approach can be described as "identify – describe – create topic map" procedure, whilst specific solutions differ. Here, we will illustrate the topic map creation process by analysing usable data sources, propose four step procedure for its creation, and show the three approaches to using the procedure.

Data Sources

In most cases, there is no need to create topic maps from scratch. Existing information resources in organizations can be used as a critical mass which will leverage the process. Topic maps, in its simplest form, can be described as a collection of Topics, Associations and Occurrences, the so called TAO of topic maps. Therefore, the most efficient information resources are those, that can be effortlessly converted into subjects, relations between them, and subjects' instances. Examples of such sources include:

- Relational databases – where primary keys describe topics, fields within one record can be occurrences, and relations help to establish associations
- Web sites – where URLs identify topics, webpage contents are occurrences, and hyperlinks show associations with other web pages
- Directory systems – where directory objects point at topics, directory schema describes topic types, objects themselves are occurrences, and associations are derived from the tree-based structure of directory
- Content management systems – which are similar to the websites, but store smaller units of data (paragraphs) and often provide more detailed descriptions
- Files in file systems – which can be treated similarly to directory systems, and again, files of specific types can contain metadata, which can afterwards be used in preparing more detailed topic map

Other way to enumerate the data sources – a more general one – would be based on source characteristics. And therefore we can identify:

- Structured knowledge – ontologies and classification systems, database schemas, document type definitions (DTDs) and XML schemas, metadata schemas
- Structured document content – with emails, newsgroup messages or accounting documents as examples
- Unstructured document content – where preparations must involve more sophisticated techniques, such as Natural Language Processing with Named Entity recognition, Concept extraction, and taxonomic classifications. Most NLP based processing tools require only raw text, therefore document transitions are not complicated

- Document metadata – analyses include properties stored in a file (such as Microsoft Office's properties, RDF-PDF or HTML Dublin Core) or externally stored properties (RDF, MPEG 21, Document Management System metadata)

Procedure

The procedure of automated topic map creation can be split into four steps. The procedure bases on a common assumption, that before creating a final map, there is a RDF (Resource Description Framework) model prepared, which is used as input data afterwards. The four steps are as follows:

1. Subject recognition
2. Information extraction and preparing
3. RDF modelling
4. Mapping RDF model into a topic map

Whilst the fourth step can be done using available tools (such as those from Ontopia), the previous three are most interesting, and let us experiment and develop methods for converting source data into a Resource Description Framework model. In this part we analyse current routines, further in the paper we will propose a method based on the two experimental concepts: term crawling and clustering hierarchy projection. Both concepts, aside from abstract considerations, are also undergoing implementation tests – sample results are presented in the text.

SUBJECT RECOGNITION

In order to identify potential topics and say something about subjects, one has to locate data sources – subject occurrences. This is highly dependent on content type. For example, in relational databases this will mean analysing a database schema and deciding, which tables contain candidate entities (one can also create his own queries based on selected tables); in document repositories the subject recognition will mean selecting a set of documents, which will be processed later on. Once we have identified the subjects and their occurrences, we should prepare unique URIs. The most common method is to use one's own registered domain name to create Universal Resource Identifiers. Example URI would then be http://www.mydomain.org/URI/apps/msword - which would identify the Microsoft's application.

INFORMATION EXTRACTION AND PREPARING

After the subjects are recognized, one has to extract data needed for processing. When analysing structured sources, such as XML documents, emails etc. this is quite obvious – one has to decide which document properties are important and, optionally, to prepare them. Preparing includes data conversion, value normalizing, splitting single values into multiple values, aggregating multiple values into single values, traversing hyperlinks to collect additional data etc. For semi-structured and unstructured data, this can be trickier, and advanced information processing techniques, such as Shallow Text Processing (Abramowicz and Piskorski 2003), Natural Language Processing (Named Entity recognition, Concept extraction, and taxonomic classification) have to be used. Further parts of the paper will show that Term Crawling and Clustering Hierarchy Projection techniques can help here as well.

RDF MODELLING

RDF models describe objects, which correspond to topics in a topic map; their properties correspond to occurrences or associations with other topics. A RDF model consists of statements (often called triples), which have three parts: subject, property, and value. Subject describes a resource, the statement is about; Property describes the property type assigned to subject; Value contains a specific value of the property of the subject. Subject, property, and value can contain URIs, value can also contain other data types, such as strings, integers etc.

MAPPING RDF MODEL INTO A TOPIC MAP

This step, performed mostly by automatic tools, involves analysing RDF triples, and deciding whether a triple describes a topic, an association, or an occurrence. Preparing topics, associations and occurrences is the final step in the topic map creation, and further activities focus only on refining the map.

Processing Approaches

The described procedure of automated topic map creation can be used in a variety of environments, using a wide spectrum of source data. Depending on expected application, one of three approaches can be chosen from: one-time processing, repeated batch processing, and continuous processing.

ONE-TIME PROCESSING

One time move from legacy system to topic map is very effective, because the legacy system is no longer used and full power of topic maps can be used from the beginning. The disadvantages include need to roll out legacy indexes or supporting users who have not rolled out, and therefore do not have access to the latest data.

REPEATED BATCH PROCESSING

The repeated batch processing can be triggered or scheduled. It allows for using existing, legacy indexes, and topic map at the same time. However this procedure is more resource consuming, less reliable (especially when source data schemas change), and does not guarantee that topic map is up to date.

CONTINUOUS PROCESSING

Continuous processing, or wrapper around existing system, is a most complex technique. It lets users use existing tools and indexes and at the same time it updates the topic map, so that it is always up to date. However, if the existing system changes, significant development efforts may be required in order to maintain operability.

The approach proposed in this article can be applied in all the approaches mentioned above. It is also feasible in the last, most complicated, case. When a document collection is changing (new documents are added), the topic map should change in order to resemble new structure. In our approach it is possible to track new vocabulary which may be introduced with new documents, and include it in one of the dictionaries used in the procedure described below (Clustering Hierarchy Projection and Term Crawling).

Selected knowledge discovery techniques

Methods of automated topic map creation, sketched in the previous part of the text (see section 0), are very efficient for structured and semi-structured data. However, when we try to apply the described procedure to unstructured data, such as collections of documents from the Web, a number of questions arise. There is no easy way to point out subjects, and associations. One, previously mentioned, way would be to use Natural Language Processing techniques, however they require significant effort to build rules for different languages (as, obviously, not only English documents may be processed), associations proposed by those techniques base only on documents' contents (and therefore overlook assumedly well known relations – contextual information).

Recent work on ontology building (which is a shared and formal specification of the vocabulary and assumptions about its use describing certain, limited reality) gives hints on which techniques can be useful when we deal with unstructured data (Maedche 2002). These techniques originate from data mining and knowledge discovery, as described in section 0. We adopted three of them to propose new approach towards topic map creation. The first one – Term Crawling – will let us automatically identify relations between concepts, the second one – Clustering Hierarchy Projection – is based on hierarchical clustering of documents, thus allowing to identify additional subjects in a document collection and hierarchical relationships. The third one – association rules discovery – is used to extend topic map with non-hierarchical associations. These methods, jointly used, provide a framework for creating RDF triples from unstructured data, and eventually creating a topic map, which would not be possible when using standard methods.

This section provides background on data mining techniques used in proposed approach to topic map creation.

Clustering

The basic idea behind clustering is that documents can be grouped according to their content similarity without any prior knowledge or assumptions concerning this content. There are various approaches to clustering as described in (Bhatia and Deogun 1998) and (Steinbach, Karypis et al. 2000). Usually the techniques are divided into K-means clustering and hierarchical clustering. The former is based on the following procedure:

1. Select initial K points among documents (each document is represented by vector of its terms frequencies) – these points are called centroids. The mathematics defines centroid of n point masses m_i located at points x_i as a centre of mass with the formula like in Eq. (1)

$$\bar{x} = \frac{\sum_{i=1}^{n} m_i x_i}{\sum_{i=1}^{n} m_i} \qquad (1)$$

   In our case the x would be terms frequencies and m – their weights.
2. Assign all remaining points to the closest centroid therefore creating clusters
3. Recompute centroid for each cluster.
4. Repeat 2 and 3 until centroids don't change.

However this approach proved to give worse clusters than the hierarchical clustering, which is usually described as below:

1. Create primary set of clusters where each document is represented by single cluster.
2. Compute similarity between all clusters using selected similarity measure – this creates similarity matrix.
3. Reduce the clusters number by merging the closest clusters.
4. Update the similarity matrix.
5. Repeat steps 3 and 4 until the desired number of clusters is reached.

There are two approaches to creating a hierarchical tree of clusters (Maedche 2002) – either bottom-up, starting with individual objects and grouping the most similar ones into clusters, or top-down, starting with one large clusters, containing all objects, and dividing such cluster into smaller ones. The three functions used are *sim*, *coh* and *split*. The *sim* function returns similarity measure between any two given documents. There is a number of similarity measures to choose from, according to the model. Here, the *cosine* measure may be a good choice. The *coh* function measures cluster coherence, and the *split* function splits one cluster into more objects.

Algorithm 1.: Hierarchical Clustering Algorithm – Bottom-up (Maedche 2002)

```
Require:    a set $X = \{x_1, \ldots, x_n\}$ of objects, n as the overall number of objects,
            a function sim: $R(X) \times R(X) \rightarrow \Re$
Ensure: the set of clusters K (or cluster hypobook)
    for i:=1 to n do
        $k_i := x_i$.
    end for
    $K := \{k_1, \ldots, k_n\}$
    j:=n+1
    while |K|>1 do
        $(k_{n1}, k_{n2}) := \arg\max_{(k_u, k_v) \in K \times K} sim(k_u, k_v)$
        $k_j = k_{n1} \cup k_{n2}$
        $K := K \setminus \{k_{n1}, k_{n2}\} \cup \{k_j\}$
        $j := j + 1$
    end while
```

Algorithm 2: Hierarchical Clustering Algorithm – Top Down (Maedche 2002)

```
Require:    a set $X = \{x_1, \ldots, x_n\}$ of objects, n as the overall number of objects,
            a function coh: $R(X) \rightarrow \Re$
            a function split: $R(X) \times R(X) \rightarrow R(X)$
$K := \{X\} (= k_1)$
j:=1
while $\exists k_i \in Ks.t. |k_u| > 1$ do
    $k_u := \arg\min_{k_v \in K} coh(k_v)$
    $(k_{j+1}, k_{j+2}) = split(k_u)$
    $K := K \setminus \{k_u\} \cup \{k_{j+1}, k_{j+2}\}$
    j:=j+2
end while
```

Hierarchical clustering comes in different flavours depending on the similarity measure taken. Average linking is based on the mean of documents vectors – the centroid (also defined as a centre of a cloud of points – in most approaches it is simply a vector of average term frequencies from all documents in the cluster). Complete linking uses the dissimilarity measure which is the greatest distance among points in compared clusters. Single linking uses dissimilarity in the opaque way – here the dissimilarity is defined as the minimum distance between any points in compared clusters. Various distance metrics are used to compute similarity or dissimilarity but the most common is the cosine measure which is dot product of the two vectors divided by their lengths product.

$$\cos(\vec{x}, \vec{y}) = \frac{\sum_{x \in X, y \in Y} xy}{\sqrt{\sum_{x \in X} x^2 \sum_{y \in Y} y^2}}$$

In a tool described in this article we also use this measure.
Hierarchical clustering is particularly interesting in our approach because the algorithm generates hierarchical tree of clusters as it merges them into bigger clusters. The Figure 1 shows the sample hierarchy generated by hierarchical clustering algorithm. The initial clusters (Cluster 1, Cluster 2 … or C1, C2 etc.) are merged during the subsequent iterations of the algorithm, based on their similarity. The bottom part of the picture shows inclusion of the primary clusters in their conglomerates.

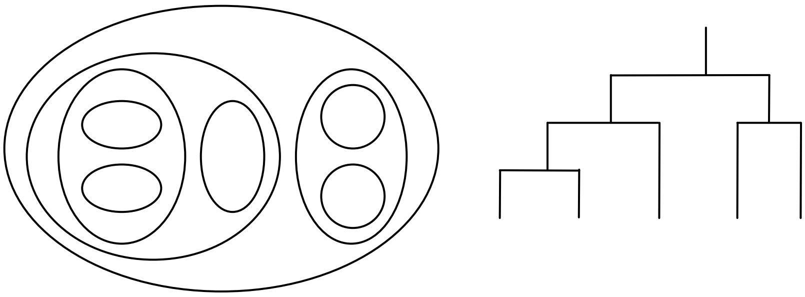

Figure 5. The hierarchy tree generated by a clustering algorithm

Term Crawling

The aim of term crawling method is to build a network of associations between terms. The semantic network is built by analysis of concurrent term presence in documents on the World Wide Web. Assuming that there is a tool for gathering information from Internet sources and assessing their relevance, term crawling aims to indicate, that even unstructured data of limited trust can be utilized in information systems.
The network of associations, created by term crawling will then be used in creating associations between clusters created by CHP (Clustering Hierarchy Projection).

1. In order to build a semantic network of associations between terms, we have to specify a starting point – this will be the primary term.
2. The primary term is used as a query, and – through Google API (Dornfest 2002) – submitted to the search engine, which replies with a list of web pages relevant to the query.
3. Web pages are downloaded
4. Value of web pages for the process is assessed – this includes length, format, and language analyses.
5. Downloaded web pages are tokenised, and tokens weights are estimated – this leads to creating n-dimensional matrix, where n is equal to number of aspects of each token (at least its URI and weight).
6. The matrix is then processed. The processing techniques are still in the experimental phase and we are looking for the most efficient technique. Currently we create intersections of term sets from each webpage and as a result, after applying stop-word list, we get a list of terms related to the primary term.
7. In that phase, the network of associations can be updated. If related terms are already contained in a network, then new associations are created, otherwise new nodes and associations are created.
8. Then, according to user's needs, recursive searches can be deployed. Breadth-first search attempt is preferred. Continuous running of term crawling mechanism leads to building large semantic networks of associations between terms. Apparently, such algorithm will result in gigantic maps, if not constrained by depth boundaries. The boundaries should be specified beforehand.

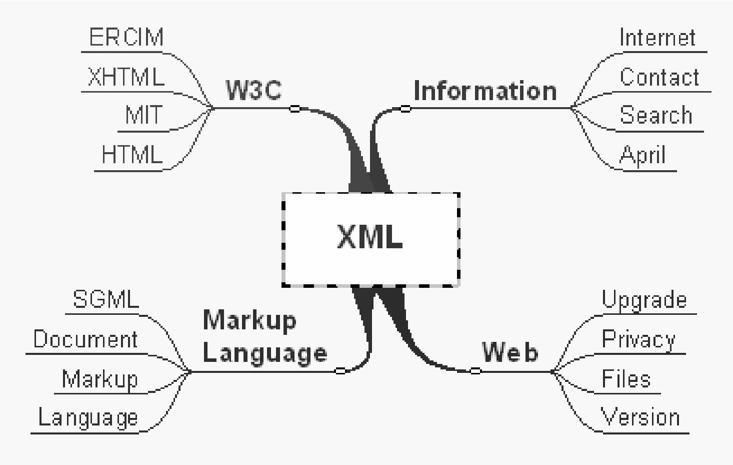

Figure 6. Network created with Term Crawler – "XML" as an initial term

Figure 2 shows a sample map created with Term Crawling algorithm, with crawling depth limited to two levels and stop list applied. As it can be observed, not all results are valuable for further processing (see "April" – a term associated with "Information"), but overall structure looks very promising.

Affinity grouping and association rules

Affinity grouping and association rules are useful for identifying coexisting facts. When analysing large document repositories, they may be used – as in case presented in this article – for identifying coexisting concepts. An effective way of identifying important rules is by representing them using first order logic. A rule may be represented as:

$$r_1(a_1,v_1)\wedge r_2(a_2,v_2)\wedge\ldots\wedge r_j(a_j,v_j)\Rightarrow r_k(a_k,v_k)\wedge r_1(a_1,v_1)\wedge\ldots\wedge r_n(a_n,v_n)$$

Where:
a_i is an attribute
v_i is a value (simple or compound)
r_i is a predicate

The left side of a rule is called rule body, the right side – rule head. The sentence above may be either true (confirming the rule, when rule body and head are both true) or false (violating the rule, when rule body is true, and head is false). Every rule has two measures: support (S) and confidence (C) which are defined below in equations (2) and (3). Rule support is the ratio of the number of objects holding the rule to the number of all objects in analysed dataset. Confidence is the ratio of the number of the objects where both rule body and head hold to the number of objects where rule body holds.

$$S=\frac{|T^H|}{|T|} \quad (2)$$

$$C=\frac{|T^H|}{|T^B|} \quad (3)$$

Where:
T is a set of all objects in the analysed dataset
T^B is a set of objects where rule body holds
T^H is a set of objects that hold the rule (which implies that $T^H \subseteq T^B$)

Topic map creation process

Topic map creation is a two faceted process – first, there should be created an initial topic map, basing on a document collection. After that, there is a place for a continuous process of updating topic maps (as described in point 0 – Processing approaches). The following part of the paper focuses on a first facet – creating an initial topic map from a document collection. We highlight six main steps of the process: preparing document collection, generating base dictionary, performing term crawling and clustering (those two subprocesses can be run in parallel), identifying associations, and preparing RDF triples for further processing.

Document Collection

The collection that we use for creating topic map is gathered from the CNN.com portal – it is a set of several hundred news stories from main categories (e.g. health, politics, travel, technology). The documents are partially described with metadata (publication date, title and category).

The most interesting results of Clustering Hierarchy Projection and Term Crawling can be obtained for collections of documents of different subjects – this is also an advantage of topic maps. Still, in order to use thesauri efficiently, the documents should belong to some general domain. Topic maps, CHP, and TC are not necessarily the best solution for document collections of highly specific domains.

Base Dictionary

The base dictionary, which contains selected words from the document collection, may be divided into domain specific dictionaries. The standard stoplist for English language is used to clear the dictionary, and typos are removed. Then the dictionary is split into smaller dictionaries (this is hand-made). The number of child dictionaries depends on the main topics that we would like to distinguish in our topic map. The most basic approach would be to divide the dictionary according to main categories found in a document collection. More sophisticated approach would be to identify topics based on geographical names or economic entity names. The dictionaries used for subsequent passes of clustering algorithm do not have to be disjoint.

However, the above remarks are only heuristics, and the final decision concerning rules of the dictionary split depends both on the collection (its specialization and range) and the desired result. Sometimes repeated experiments are necessary to obtain satisfying results. In fact, expert's skills are required to asses the aspects of the collection that should be identified and used in the procedure. This is a potential subject of future publications.

Term Crawling

After the base dictionary has been created, term crawling mechanism is used to identify and store relations between terms. Each term is taken from the dictionary, and related terms are searched. Depth of term crawling should be set manually, but two levels (as in Figure 2) seem to be efficient for most applications. The term crawler filters only stop words – association network may contain non-dictionary terms as well. As an outcome of term crawling, there will be created a network with N vertices (where N is a number of terms in dictionary incremented by other related terms found out by TC) and M edges (term relations, as indicated by TC). This network can be used to identify associations in the topic map (see below).

The term crawler is implemented in Visual Basic .Net and uses extensively Google API, based on Web Services. The semantic network, created by term crawler may be stored in XML files or relational database. This is out of the scope of this paper, but it is worth mentioning, that the semantic structure can be successfully used for user query modification (broadening queries and including related terms), which can be of a great value for e-business. Catalogue browsers can for example propose products that are in some way related to those specified by customer (moped instead of a motorcycle etc.). The semantic network can be also treated as a preview of present-day associations between subjects, with respect to their evolution (which can also be represented in the network).

Clustering Hierarchy Projection

Clustering proved to be valuable in dividing large document collections into smaller, meaningful parts (i.e. clusters represent certain topics). The rationale behind our approach is to use clustering to assign certain topics, created by the clustering algorithm, to clustered documents. We use the hierarchy of generated clusters to create hierarchy of topics and their occurrences – the documents. Except from using hierarchy of clusters (which is usually discarded in clustering applications) we use several dictionaries in the algorithm. Classical approach to the clustering uses single cleared dictionary. The stoplist is used to eliminate junk-words and in some approaches the words are stemmed (according to the algorithm first described in (Porter 1980).

In our approach we split this dictionary to obtain smaller dictionaries for specific aspects of a domain that the documents regard. The clustering of the same documents set using different dictionaries produces different hierarchies of clusters. This can be intuitively viewed as a projection of n-dimensional term space (where n is the number of terms in the whole dictionary) to a set of m-dimensional spaces (where m<n). One could argue that we loose some information because clustering is done with smaller dictionary. But in exchange we gain a general view on the document collection and we can build the topic map based on several hierarchies and associations between them, which emerge from co-occurrences of the same documents in each hierarchy.

We use bottom-up hierarchical algorithm with each dictionary that has been selected. Every iteration of the algorithm is logged. This information is used to create the hierarchies of clusters. The hierarchies may differ in depth, depending on the technique used to create the tree. The sample tool allows for adjusting that. For example, when merging two clusters of similar size (measured in number of documents in each cluster) they form a new, more general cluster. On the other hand, merging single-document cluster with a cluster containing several documents does not produce a new cluster. The single-document one is joined with the bigger one.

Cluster can be characterized by a set of terms that appear in its documents and in the dictionary used for clustering. These terms can be ranked according to their frequency in the whole cluster or according to their input to the similarity of the documents in the cluster (the smaller is a distance between documents computed using single term frequency, the more similar documents are and therefore the better given term describes the whole cluster).

The most popular measure for weighting terms is *term frequency – inverse document frequency* measure, denoted $tfidf_{t,d}$ and defined in equation (4). This measure is used throughout the topic map creation process and therefore is readily available for choosing terms that describe clusters best.

$$tfidf_{t,d} = \frac{freq_{t,d}}{|d|} * \log(\frac{|D|}{df_t}) \qquad (4)$$

Where:

$\text{tfidf}_{t,d}$	is tfidf measure of term *t* for document *d*
$\text{freq}_{t,d}$	is the number of occurrences of term *t* in document *d*
|d|	is the number of all terms in document *d*
|D|	is the number of documents in the document collection
df_t	is the number of documents that term *t* occurs in.

In current stage these terms (with the highest tfidf) are used by human author of a topic map to give the topic name. Automatic assignment of a topic name can be achieved using thesaurus which is discussed in section 0 of this paper.

Clustering also provides linking topics to occurrences – the documents in a document collection. Such link has a form of URI pointing at a given document. Although, using our technique this is achieved in an obvious way, this becomes a non-trivial task when creating a topic map without automating tools. Identifying all occurrences of a given topic in an enormous collection may be challenging for a human author, but is easily achieved (with certain degree of accuracy) by a retrieval machine.

The outcomes of the clustering mechanism are several hierarchies of clusters that can be mapped to topics in an arising topic map. The next step is to create or interpret the existing associations.

Associations

The preceding procedure produces data structures that suggest existence of certain associations between topics. These associations can be divided into:

1. *Generalizing* – these associations are derived from hierarchies generated during the execution of clustering algorithm. Merging two clusters into one more general allows creating association of a type: *generalization*, which connects smaller cluster with the bigger and more general one.
2. *Specifying* – these are the opposite of the "generalization" associations, but they are created from more general to more specific clusters, down the hierarchy tree.
3. *Unnamed* associations between different hierarchies – these are based on the co-occurrence of the same documents in the clusters belonging to the hierarchies created with different dictionaries. The hierarchy trees are compared and the list of suggested associations is created. The list is created in the following manner: the more common documents appear in compared clusters, the stronger the association is assumed to be, and therefore it appears higher on the list.

   Affinity grouping and association discovery techniques (described in paragraph 0) are used to identify unnamed associations. To find associations between two clustering hierarchies generated with different dictionaries and thus between separate topics one has to apply the following procedure:

   - for each document in both hierarchies get clusters that the document belongs to – they form rules' body (of course the bottom clusters containing single documents are not taken into account, as well as top cluster – if it contains the whole document collection)
   - find clusters in the second hierarchy that the document belongs to – they form rules' heads
   - given set of objects generated above select unique set of rules and compute support and confidence measures for them

   Rules with high confidence measure are particularly useful since they indicate strong relationship between topics from separate hierarchies. It is crucial to take into account relationship's direction and apply the above procedure in both directions analysing both hierarchies as a source for rules' bodies. This provides that associations of clusters with similar documents and documents number are found to be the strongest.

   The ordered (according to confidence) set of associations is then presented to the topic map engineer, who can discard an association, if he considers it unnecessary, or give it a name. For example if one hierarchy was obtained with the dictionary containing corporations' names and the other with the dictionary containing product names, the association name could be "is a product of".
4. *Unnamed* associations between clusters as indicated by term crawling – each of clusters is described with a term (or set of terms). Those terms, further converted into topics in a topic map, are compared with a network created by term crawling mechanism. The comparison lets us identify even more relations between clusters – depth of association network searching can be decided upon each time a topic map is generated. Association discovery using support and confidence measures can be performed with term crawling if the

procedure described in paragraph 0 is to be modified. Instead of taking into account only these terms that appear in all the analysed documents, one could measure the number of occurrences of a certain term – this could be used to compute the confidence measure for the rule: query term → found term.

Similarly to unnamed associations between different hierarchies, unnamed associations between clusters produce a list of suggested relations, which can be processed by human authors afterwards.

RDF Triples

The Resource Description Framework Model and Syntax Specification defines the RDF data model, and basic serialization syntax. It became a W3C (World Wide Web Consortium) recommendation in 1999. The data model is basically a directed graph. Its elements include entities and binary relationships. The relationship is represented by RDF statement (also called RDF triple). A statement can be represented by two nodes and a directed arc between them:

- Subject – the resource the statement is about (URI)
- Property – the property, being assigned to the subject (URI)
- Value – value assigned to the property (URI or string literal)

In our case, most popular RDF triples would be for example:

- (http://my.org/news/1, #topic, http://my.org/thesauriitems/54) – representing occurrence of a specific term
- (http://my.org/thesauriitems/23, #generalizing, http://my.org/thesauriitems/66) – representing generalization of topics (terms)
- (http://my.org/thesauriitems/66, #specifying, http://my.org/thesauriitems/23) – representing specification of topics (terms)
- (http://my.org/thesauriitems/38, #suggested, http://my.org/thesauriitems/95) – representing suggested association between terms, based on term crawling, and identifying associations based on coexistence of documents in different clusters (as described above)

After creating RDF triples, the process of creating a topic map becomes obvious, and existing tools can be used for that purpose.

Future work

The proposed solution, although not fully automating the topic map creation process, has proven to be useful when creating a topic map describing a document collection. The approach could be further enhanced in several areas. There is significant effort towards identifying topics in plain text files. These topics may be both single words or short phrases and whole text parts or chapters devoted to specific subject. Identifying topics is not enough, one have to name them. Thesauri structures can be used to support it at a larger scale. These structures contain information about words synonyms, hierarchies of meaning and phrases, which may be useful when naming topics.

A serious effort in the information retrieval and document understanding is aimed at automatic topic recognition in texts. Here, the word topic stands for a larger part of a given text, devoted to some specific subject. Such techniques may be helpful when identifying topics for topic maps. Attempts have been made to use the described method to support continuous processing of the document collection. As mentioned above it involves modifying the topic map as new documents (potentially significantly different from others in the collection) appear. The approach involves creating new dictionary from the new terms found and adding topic hierarchy generated to the already existing. Some associations can be provided by term crawling, but further research in this field is necessary.

The most troublesome on this level of generality is association creation and interpretation. In the existing solutions associations are created based on examined language structures. In our approach a number of general associations is created, however naming the unnamed ones is currently not solved in our solution.

The applications of topic maps in the domain of information systems development demands further research. The insight into unstructured data or text semantics that may be obtained using automatic topic map generation can be useful for system design, as pointed in the Introduction. However this requires further improvement both concepts and tools.

REFERENCES

Abramowicz, W. and J. Piskorski, 2003, "Information Extraction from Free-Text Business Documents." **Effective Databases for Text & Document Management 2003**: 12-23.

Abramowicz, W., Kowalkiewicz, M., and Zawadzki, P., 2002, Tell me what you know or I'll tell you what you know. Skill map ontology for information technology courseware, Mehdi Khosrow-Pour (ed.), **Issues and Trends of Information Technology Management in Contemporary Organizations,** Information Resources Management Association International Conference, Seattle, USA, 2002, Information Science Publishing.

Abramowicz, W., Kowalkiewicz, M., and Zawadzki, P., 2003, Ontology Frames for IT Courseware Representation. In E. Coakes (Ed.), **Knowledge Management: Current Issues and Challenges**. IRM Press

Ahmed, K., 2000, "**Topic maps for repositories**." XML Europe Conference.

Andersen, S. Q., 2003, forskning.no. 2003.

Baeza-Yates, R., and Ribeiro-Neto, B., 1999, **Modern Information Retrieval** ACM Press, Addison Wesley Longman Limited, USA

Berry, M. J. A. and G. Linoff, 1997, **Data Mining Techniques for Marketing, Sales and Customer Support**. New York, John Wiley & Sons Inc.

Berry, M. J. A. and G. Linoff, 2000**, Mastering Data Mining. The Art and Science of Customer Relationship Management**. New York, John Wiley & Sons Inc.

Bhatia, S. K. and J. S. Deogun, 1998, "Conceptual clustering in information retrieval." **IEEE Transactions on Systems, Man and Cybernetics**: 427-436.

Ding, C., and He, X., 2002, Cluster merging and splitting in hierarchical clustering algorithms, **2002 IEEE International Conference on Data Mining**, Maebashi, Japan

Dornfest, R., 2002, Google Web API, The O'Reilly Network.

Gómez-Pérez, A., 1999, Evaluation of taxonomic knowledge in ontologies and knowledge bases, proc. of the **Knowledge Acquisition Workshop**

Grønmo, G. O., 2000, Creating semantically valid topic maps, **XML Europe Conference**, Paris, France

Grønmo, G. O., Automagic topic maps, Retrieved April 26, 2003 from: http://www.ontopia.net/topicmaps/materials/automagic.html

Gruber, T. R., 1993, Toward principles for the design of ontologies used for knowledge sharing, **International Workshop on Formal Ontology**, Padova, Italy

ISO, 2000, **Information technology - SGML applications - topic maps**. ISO/IEC 13250. Geneva.

Knight, J. R., 1996, Discrete Pattern Matching Over Sequences and Interval Sets, Ph.D. Dissertation, Department of Computer Science, The University of Arizona

Ksiezyk, R., 2000, Answer is just a question [of matching topic maps], **XML Europe Conference**, Paris, France

Maedche, A., 2002, **Ontology learning for the Semantic Web**. Boston, Kluwer Academic Publishers.

Moore, G., 2001, RDF and Topic Maps – An Exercise in Convergence, Retrieved April 26, 2003 from: http://www.topicmaps.com/topicmapsrdf.pdf

Oommen, B. J., and de St. Croix E. V., 1994, String taxonomy using learning automata, **IEEETSMC: IEEE Transactions on Systems, Man, and Cybernetics**

Pepper, S. 2000. "The TAO of Topic Maps." **XML Europe Conference**.

Pepper, S., 2002a, Ten Theses on Topic Maps and RDF, Retrieved April 26, 2003 from: http://www.ontopia.net/topicmaps/materials/rdf.html

Pepper. S., 2002b, The Ontopia MapMaker, Retrieved April 26, 2003 from: http://www.ontopia.net/topicmaps/materials/MapMaker_files/frame.htm

Porter, M. F., 1980, "An alogrithm for suffix stripping."

Resource Description Framework (RDF) Model and Syntax Specification, Feb. 1999. W3C Recommendation.

Rittershofer, A., 2002, Lernen mit Topic Maps. 2003.

Sowa, J. F., 2000, **Knowledge Representation: Logical, Philosophical, and Computational Foundations**, Brooks Cole Publishing Co., Pacific Grove, CA

Steinbach, M., G. Karypis and K. V., 2000, **A Comparison of document clustering techniques**. 2003.

Walsh, N. and L. Muellner, 1999, **DocBook: The definitive guide**, O'Reilly & Associates.

Wrightson, A., 2001, Topic Maps and knowledge representation, Retrieved April 26, 2003 from: http://www.ontopia.net/topicmaps/materials/kr-tm.html

Zhao, Y., and, Karypis, G., Evaluation of hierarchical clustering algorithms for document datasets", Retrieved April 2003 from: http://citeseer.nj.nec.com/zhao02evaluation.html