Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

The Classification of Structured Knowledge Objects[1]

Guy W. Mineau
Department of Computer Science
Université Laval
Québec, Québec
Canada, G1K 7P4
mineau@ift.ulaval.ca

This article presents the automation of a classification method applicable on graph-described knowledge objects. We need graphs to describe knowledge structures in a computer, but they imply a very heavy computational overhead in order to be compared and classified. Simplification heuristics have to be enforced. Such heuristics imply that the classification structure becomes less semantically significant since less analysis is actually done in order to produce it. This paper deals with the problems related to graph classification, and seeks solutions to that effect. Thus it advocates synergism among the different aspects of classification research.

1. INTRODUCTION

Classification processes are very important in human behaviour. They take part in different learning activities, are useful in memory organization, and help to create a model of a reality (Martinez & Kesner, 1991; Puff, 1979). Thus, they also play an important role when knowledge is processed by a computer, especially when learning capabilities are implemented within a computer (Lebowitz, 1986). The problems related to their automation then need to be addressed.

In computer science, a subfield of artificial intelligence (AI), called machine learning, addresses the problems related to the automation of learning capabilities (Michalski, Carbonell & Mitchell, 1983; Michalski, Carbonell & Mitchell, 1986; Kodratoff & Michalski, 1990). Machine learning covers different learning paradigms, such as learning by being told, by analogy, by induction, by discovery, by example, etc. These paradigms all rely on some classification procedure in order to either infer new knowledge, or to organize the available knowledge. In both cases, some classification mechanism is needed in order to improve the learning capability of the system, which in turn, should improve the overall behaviour of the system. Consequently, one of the major research issues in machine learning is the classification process underlying any learning capability.

Naturally, the replication of human thought requires the use of some representation language so that knowledge structures can be represented and processed inside a computer. For that purpose, knowledge representation formalisms have been developed in AI. They all offer a certain degree of expressiveness. Those that are the most expressive represent knowledge structures as graphs, or as structures that can be related to graphs. These graph structures are called *semantic networks*.

---

1. This research is sponsored by the Natural Sciences and Engineering Research Council of Canada, under grant #OPG0105365, and by the FCAR Research Council of the Province of Quebec, under grant #94-NC-0944.

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

In a semantic network, the vertices represent *concepts*, i.e., objects of the reality to be expressed. They can be concrete concepts, like "elephant", "dog", "cat", "table", etc., or they can be abstract concepts, like "love" (the noun), "love" (the verb), "hate" (the noun), "feel", "redness", etc. In a semantic network, neighboring vertices are related through qualified links which represent semantical relations between them. For instance, "on", "under", "loves", are such relations. It is often the case that we encounter relations which are directly mapped from some sentence representation mechanism such as Fillmore's case grammar (Dirven & Radden, 1987). In that case, we would have the following relations: "agent", "object", "recipient", "location", etc. We could add other relations as well, such as spatial relations like "on", "under", "left-of", "above", "right-of", etc., and such as temporal relations like "before", "after", "during", "past", etc. Also, there is always some reference mechanism available in order to relate the concepts to existing or non-existing objects in the reality to be modelled. For instance, we would distinguish between a cat, the cat, and Morris the cat. Consequently, we could express the following sentence: a cat feels love for Morris the cat which is on the table, by the graph of Figure 1. In this graph, boxes represent concepts (the vertices), and circles represent relations (qualified links) between concepts. In this example, we used Sowa's conceptual graph formalism for representing knowledge (Sowa, 1984).

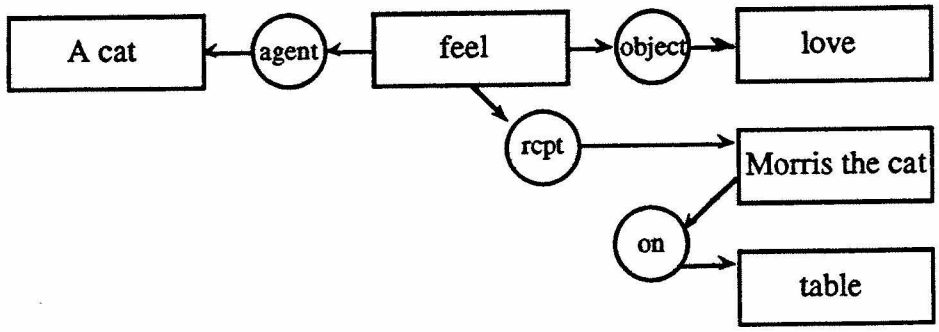

Figure 1: The representation of a sentence

In Figure 1, the "feels love" part of the graph could have been represented directly using the "loves" relation. Representational issues such as the set of concepts and relations necessary to model the domain of semantics, the available representational alternatives, and the preferred representational choices, are established by the knowledge engineer after a thorough analysis of the domain to be modelled. Such issues are related to the ontology formation, knowledge modelling, and knowledge acquisition problems (Guarino & Poli, 1993; Hart, 1986; Sowa, 1993).

Other features are also available to represent more complex objects like sets, modalities (desires, thoughts, hypothetical worlds), situations, events, plurals, etc. A thorough examination of semantic

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

networks formalisms is outside the scope of this article. We will refer the reader to (Sowa, 1984; Sowa, 1991) for more details.

When a knowledge structure is represented using a particular knowledge representation formalism, and when it encodes the semantics relevant to the description of a particular thema, object, concise thought or sentence, it is called a *knowledge object*. When a knowledge object is represented using a semantic network formalism, it is called a *structured object*. In computer science, a repository of knowledge objects is called a *knowledge base*, in analogy to repositories of data, called *databases*.

The classification of the structured objects composing a knowledge base is the aim of the research presented in this paper. As we know, classification mechanisms produce a set of classes, also called *clusters*, which represent the different similarities detected among subsets of the objects to classify. Depending on the requirements and nature of the application domain, these classes could be hierarchically related, could partition the set of objects, could be overlapping, and so on. Different classification mechanisms have different goals. Our goal is to produce a classification structure which allows the representation of all similarities detected among subsets of knowledge objects. We want to maximize the number of groupings that the classification process can detect. Basically, we seek to implement a learning by discovery mechanism on top of a knowledge base; that is, we want the system to be able to discover unsuspected clusters of objects. This information will prove to be useful when analyzing the domain of semantics modelled by the knowledge base, and when querying the knowledge base in order to provide the user with relevant and complete information for his or her needs.

In effect, a classification structure built over a knowledge base could be beneficial for diverse inferential and retrieval processes. A classification structure regroups objects sharing common semantics: it creates classes of similar objects. Queries about classes of objects can be answered more effectively since information about these classes has been explicity stored within the classification structure. Also, a classification structure produces a partial order relation of generality (and specificity) induced over all classes (as explicited in section 2). When explicitly stored, this relation helps in the exploration of the classification structure, and thus, of the underlying knowledge domain. This is used in different analysis and retrieval activities in diverse application areas. We used it for domain analysis in software development (Mineau, Godin & Missaoui, 1993a), knowledge acquisition (Mineau, Godin & Missaoui, 1993b) and ontology integration (Mineau, 1993b). Please refer to (Mineau, 1990; Mineau & Godin, 1993) for more details.

Since we deal with knowledge structures which represent symbolic data, as opposed to numerical data normally processed by computers, the classification processes applicable to that type of data are called *conceptual clustering techniques*. Conceptual clustering is a subfield of machine learning, and first emerged from numerical clustering research (Michalski, Stepp & Diday, 1981). The following section treats of conceptual clustering and of its major pitfalls.

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

2. CONCEPTUAL CLUSTERING

In the literature, two main streams of clustering techniques can be found: numerical and conceptual. The principal advantage of conceptual clustering over conventional numerical clustering, also called *numerical taxonomy* (Jain & Dubes, 1988), is the fact that in addition to producing a classification scheme (a set of hierarchically related classes), conceptual clustering also associates with each class a *symbolic description* which gives an interpretation of the intent of the class (Kodratoff, 1988). This description is called *characteristic description.* Conceptual clustering methods are strongly oriented towards symbolic data as opposed to numerical data although some methods integrate both types of data. Class descriptions can therefore be used to determine the common properties of the objects of the class. More formally the conceptual clustering problem can be specified by the following (Michalski & Stepp, 1983):

- Input: given a set of instances (objects) with their associated descriptions;
- Output: find a *concept hierarchy* (classification structure) made of,
    - a set of clusters (classes) of instances;
    - a characteristic description for each cluster;
    - a hierarchical organization of the clusters.

A concept hierarchy is a set of hierarchically related classes, based on the subsumption of the descriptions of the classes. Since each class represents a subset of similar objects, and since different subsets can be related through a partial order of inclusion, the classes themselves can be organized through the same partial order relation of inclusion. So we can define the $\leq$ operator which states that if $c_1 \leq c_2$, then c_1 is said to be more specific than c_2 (or c_2 is said to be more general than c_1). In other terms, c_1 represents a subset of objects contained in the subset of objects represented by c_2. Consequently, the description of c_1 will implicitly imply the description of class c_2. The latter description is said to subsume the former. In that case, class c_1 is said to be a *child* of c_2, and c_2, a parent of c_1. A class may have many children, but depending on the characteristic of the clustering method, it may have a restricted or an unrestricted number of parents. Classification structures where the number of parents of a class is limited to one are called *trees*; others where multiple parents are allowed are called *hierarchies*. Our method aims at maximizing the discovery of all possible classes, so it produces hierarchies. In both cases, the classification structure explicitly represents the partial order of inclusion (and generality) among the different classes, and in mathematical terms, is called *poset* (partially ordered set).

In a concept hierarchy, we can partially elaborate the characteristic description of a class from the descriptions of the classes which subsume it. Requiring this information only when a characteristic description is computed would minimize the amount of information replicated in different classes, and would save storage memory, which is also a limited computing resource. The resulting classification structure would be called an *inheritance network* or *inheritance hierarchy,* since the description of a class will be computed only after the relevant information from its parents is inherited by it. Of course, the inheritance mechanism in classification structures (in inheritance hierarchies) where multiple parents are allowed will consume more computing time, because more information needs to be accessed and integrated. All sorts of hypotheses can be put forth in order

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

to lessen this problem. Mineau & Godin (1993) introduce one such heuristic, as shortly presented below.

But the main problem we have to deal with is the similarity detection procedure upon which is based the class formation process. Since structured objects are described by graph structures, similarity detection among these objects is based on the use of a graph-matching technique. As we all know, this is an intractable problem which, in terms of complexity, is *NP-complete* (Harel, 1987). This means that the computing resources needed to compute similarity between graphs are insufficient to process reasonable-sized knowledge bases. The possibilities of finding similarities between graph-described knowledge objects is so overwhelming, that only a small number of objects can be considered for classification. This is in direct contradiction with our main objective of applicability on knowledge bases. Consequently, heuristics need to be applied to cut down drastically in the computing time needed by the similarity detection procedure. Of course, the resulting classification structure will probably be less significant than others built under infinite computing time. Once again, heuristics need to be used in order for very simple similarity detection procedures to maximize the significance of the similarity which is detected among the objects. Mineau & Godin (1993) present a similarity detection procedure which is solely based on syntactical resemblance of the objects. Such a procedure can be very efficient, but will be reliable only if similar objects (objects encoding similar semantics) are encoded using similar syntactical forms. Because this characteristic depends upon the nature of the application domain itself, the method identifies particular classses of application domains (those where this heuristic holds, at least partly). However, there are ways of improving syntactical similarity among structured objects, as shown in (Mineau, 1992; Mineau, 1993a).

In summary, the expressiveness of the representation language introduces a complexity factor so considerable that it hinders the applicability of any classification method on structured objects. It limits their scope up to a point where they can not be applied to a knowledge base because of its size, even though it would be beneficial to do so. Similarity detection procedures (class formation procedures) must be very efficient, thus, very simple. Section 3 below introduces our method which proposes a syntactical similarity detection procedure as the basis for class formation. The applicability of this method depends on the following two heuristics: 1) the *structural similarity heuristic*, which states that semantically similar objects will be encoded using similar syntactical forms, and 2) the *realistic size heuristic*, which states that, in practical situations, the size of any graph describing some object of the domain is bounded by a constant.

3. CLASSIFYING OBJECTS BASED ON THEIR SYNTACTICAL SIMILARITY

We propose a conceptual clustering method which can be applied to arbitrary bipartite graph structures, and which aims at classifying large sets of structured objects. In order to achieve these goals which represent improvements over previous methods dealing with structured objects (Lebowitz, 1986; Stepp & Michalski, 1986), compromises had to be made. Mainly, we will suppose that semantical similarity between the objects we want to classify can be detected by a syntactical similarity detection procedure. Consequently, the knowledge domains to which our method will apply must qualify for the application of the underlying structural similarity heuristic. Since the relevancy of this heuristic depends greatly on the nature of the knowledge domain itself,

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

our method is not universal, but when it is relevant, it can be applied to large sets of objects, i.e., to vast domains. Experimentation has shown its applicability on certain domains, mainly in information retrieval (Mineau & Godin, 1992). For particular domains, the method that we propose is both feasible and useful. These domains are those which show a natural propensity toward the structural similarity heuristic. However, as said before, ways of improving the relevancy of the heuristic have been investigated successfully (Mineau, 1992; Mineau, 1993a).

Basically, this heuristic ensures that a separation between the class formation and class description processes, which improves the efficiency of the classification algorithm, can be done without jeopardizing too much the semantics of the resulting classification structure. Similarity detection between knowledge objects can then be reduced to the detection of common subgraphs among object descriptions, which can be done very efficiently using fast index structures. This is the reason why our method is successful on large sets of knowledge objects. Characteristic descriptions of classes can be generated from these common subgraphs later on, when needed by the application. Computing resources can concentrate on one such particular task at that particular moment.

Subsections 3.1, 3.2 and 3.3 below summarize the computer implementation of the method that we propose. Subsection 3.1 introduces the internal representation for our graph structures; subsection 3.2 presents the similarity detection procedure; subsection 3.3 summarizes how the classification structure is built.

3.1. The Internal Representation of Structured Objects

Each graph *u* is broken down into a set of *triplets* $\{<c_1,r,c_2>\}$, where there is one triplet for each relation found in *u*. Each triplet $<c_1,r,c_2>$ stands for a concept c_1 which is linked to another concept c2 through an oriented binary relation r. Provided that each concept appears only once in a definition, and that all relations are binary, a graph can always be represented as a set of triplets. With conceptual graphs, concepts appear only once in a graph, and n-ary relations can be represented as n+1 binary relations. This simpler representation format is generally more appropriate for the learning algorithms used in clustering methods. For example, the graph of Figure 1 would be represented by the following set of triplets, shown in Figure 2.

{ <feel, agnt, a cat>,
<feel, object, love>,
<feel, recipient, Morris the cat>,
<Morris the cat, on, table> }

Figure 2. The set of triplets representing the graph of Figure 1.

Our method finds the common subgraph between graphs which are represented by sets of triplets. It may happen that common subgraphs include parts of triplets. For example, the following sentence: A cat feels hate for Morris the cat, would be represented by the graph of Figure 3, and the set of triplets of Figure 4.

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

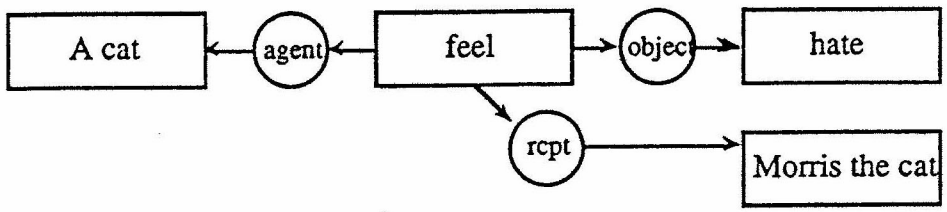

Figure 3: The representation of a similar sentence.

{ <feel, agnt, a cat>,
<feel, object, hate>,
<feel, recipient, Morris the cat>,

Figure 4: The set of triplets representing the graph of Figure 3.

It is obvious from this example, that in both cases, a cat feels some kind of emotion towards Morris the cat. The common subgraph depicted in Figure 5, would be described by the set of triplets of Figure 6.

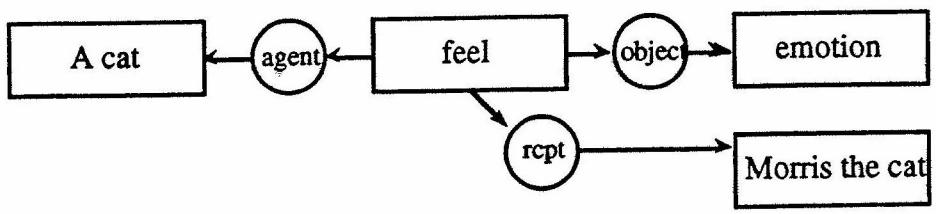

Figure 5: The common subgraph found from Figures 1 and 3.

{ <feel, agnt, a cat>,
<feel, object, **emotion**>,
<feel, recipient, Morris the cat>,

Figure 6: The set of triplets representing the graph of Figure 5.

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

In this example, "emotion" is a unifyer term for "love" and "hate". It could be found if the vocabulary is hierarchically organized, as is often the case with knowledge-based systems. However, it would be simpler not to unify "love" and "hate" in order to speed up the common subgraph detection process. Hence, the simpler set of triplets which would actually be computed by our method is represented in Figure 7, where the discording terms ("love" and "hate") have been replaced by a question mark representing an instantiation variable; the corresponding graph representation is shown in Figure 8. Postponing the instantiation of the ? variable after the classification structure is built helps to speed up the common subgraph detection process, and dissociates the classification structure from the vocabulary hierarchy normally used to find unifyer terms.

{ <feel, agnt, a cat>,
<feel, object, ?>,
<feel, recipient, Morris the cat>,

Figure 7: The simpler set of triplets representing the common graph of Figure 5.

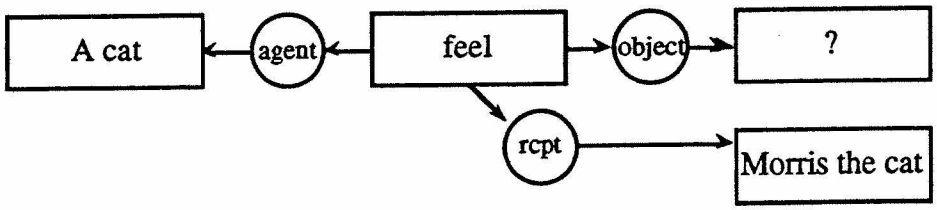

Figure 8: The graph corresponding to the object described in Figure 7.

So, from each triplet, we will generate seven other triplets, which can be seen as generalizations of the former. For instance, the generalization of triplet $<c_1,r,c_2>$ would produce the following triplets: $<c_1,r,?>$, $<c_1,?,c_2>$, $<?,r,c_2>$, $<c_1,?,?>$, $<?,?,c_2>$, $<?,r,?>$, and $<?,?,?>$. Consequently, from each set of triplets, we obtain an extended set of triplets called the *generalization set* of the object. The syntactical generalization of a triplet is done independently from the generalization of any other triplet, either in the same set of triplets or in different sets. As an example, Figure 9 shows part of the generalization set of the graph of Figure 1.

Such sets will not be as large as expected since many overlaps between generalized triplets of the same set will occur while generalizing the set. For instance, the <?,?,?> triplet will be generated from each original triplet and will not be repeated. Also in the example of Figure 9, the <feel, ?, ?> triplet will only appear once in the generalization set, though generated from the <feel, agent, a cat> and <feel, object, love> triplets. Experiments have shown that generalization sets are 30% smaller than their theoretical size (Mineau, 1990).

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

{ <feel, agnt, a cat>,
<feel, agnt, ?>
<feel, ?, a cat>
<?, agnt, a cat>
<feel, ?, ?>
<?, ?, a cat>
<?, agnt, ?>
<?, ?, ?>
<feel, object, love>,
<feel, object ?>,
<feel, ?, love>,
<?, object, love>,
<?, ?, love>,
<?, object ?>
... }

Figure 9: Part of the generalized set of triplets representing the graph of Figure 1.

3.2. The Extraction of Common Subgraphs

Common subgraphs are expressed as common triplets from different generalization sets. For instance, we would find the <feel, object, ?> triplet in the generalized sets describing the graphs of Figures 1 and 3. In Figure 1, this triplet was generated from the <feel, object, love> triplet, in Figure 3, from the <feel, object, hate> triplet. Consequently, the common subgraph of a subset of objects, is computed as the intersection of their generalized sets of triplets. Of course, not every triplet in this intersection is useful; some are redundant and can be pruned without the loss of information (see (Mineau, Gecsei & Godin, 1990) for details).

In order to compute intersections of generalized sets of triplets, any indexing scheme can be used. Building an index on the triplets themselves, where the objects being defined are kept as references with each entry (triplet) of the index, produces a data structure that we call *intersection matrix*. This matrix explicitly stores the information on what triplets are part of which object's description. Consequently, the scanning of the matrix reveals subsets of objects sharing the same triplets as part of their description. For each such subset of objects, a corresponding class C will be formed. The characteristic description for C will be inferred, when needed, from the generalized triplets belonging to the intersection of the generalized sets of triplets of the objects members of class C. These triplets will be gathered from C itself, since the generalized triplets belonging exclusively to a class will be stored within that class, and from the parents of C, since the classification structure is an inheritance hierarchy. The triplets belonging exclusively to C are those which required the creation of C while scanning the intersection matrix.

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

3.3. Creation of the Classification Structure

As seen in the previous subsection, the classes of the classification structure are created while the intersection matrix is scanned. Each of these classes is represented by a node which will be inserted into the classification structure. This insertion will be done according to the partial order of generality among the classes, as defined in section 2. This relation will explicitly be represented by links from parent to child classes. The resulting classification structure, which is an inheritance hierarchy, is called *knowledge space.*

Under the realistic size heuristic, the creation of all nodes follows a linear growth according to N, the total number of objects in the knowledge base. The creation of links between related classes follows a N^2 growth function. Experiments have shown that it is realitic to apply our method on a few thousand objects depending on the actual computer platform used (Mineau & Godin, 1992).

4. MAIN PROBLEM AND FUTURE DEVELOPMENTS

Of course, a syntactical method produces a lot of classes. Some of them do not carry significant semantics: they only represent syntactical similarity between a subset of knowledge objects. For instance, there is a syntactical similarity between the graphs of Figure 1 and 10. This similarity is expressed by the graph of Figure 11.

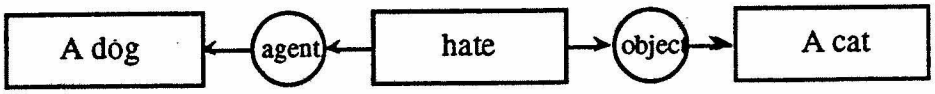

Figure 10: An example of a syntactically similar sentence.

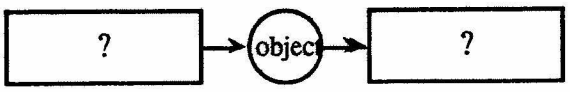

Figure 11: An unsignificant syntactical similarity.

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

It is obvious that the information carried by the class whose description is given by the graph of Figure 11, will not be helpful because it carries very little semantics. The corresponding class should not be formed. Such classes are called *useless classes*, and could be eliminated after a semi-automatic analysis of the classification structure is carried out. This pruning would simplify the classification structure, and would enhance the semantical validity of our method. It is called *class filtering*. However, useless classes help our method to be *incremental,* which is important in dynamic environments where the evolution of the knowledge base is unavoidable. This means that newly acquired knowledge objects could be added to the classification structure without having to recompute the whole structure from scratch. Since useless classes do not affect the semantics carried by other classes in any way, their presence is not troublesome when using the classification structure. However, for many application domains where the classification structure is used in order to analyze and explore the knowledge domain, they may mislead the analysis or slow it down. It would be best if all classes were semantically significant according to the domain of semantics modelled.

Consequently, ways of automatically detecting semantically significant classes is our next research consideration. We definitely need to couple our method with a pruning algorithm which would help to identify the semantically useful classes. Different approaches can be envisaged: using background knowledge, asking the user, using automatic feedback on class utility (when available), etc. We hope that class filtering will improve the overall semantical validity of the whole classification structure. Toward that objective, we need to study how class filtering is done by humans, and how it could be automated.

5. CONCLUSION

This article stressed the fact that classification processes are useful for many cognitive activities, and that their automation is thus required when implementing learning capabilities in a computer. The knowledge structures processed by these cognitive processes show a high degree of expressiveness, up to a point where their encoding requires a graph-based formalism. This introduces computational complexity which calls for draconian heuristics in order to make their classification possible. In effect, when applied to large knowledge domains, classification algorithms face the obvious conflict between efficiency and semantical validity of the resulting classification structure. Trade-offs have to be made. Their scope and impact need to be assessed. Our past and current research deals with these issues.

A conceptual clustering method applicable on graph-described knowledge structures was designed according to these restrictions. It produces a classification structure which is called knowledge space. Our method was briefly presented in this article. It was demonstrated elsewhere that it was both feasible and useful (Mineau, 1990).

However, its main problem is the detection of semantically significant classes. We desperately need to search for ways of efficiently validating the semantics carried by each class; thus, we need to prune the resulting classification structure. Synergism being a part of scientific discovery, this is where other research projects in different aspects of classification research could be of tremendous value to our own research agenda.

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

6. REFERENCES

Dirven, R. & Radden, G. (Ed.). (1987). *Fillmore's case grammar: a reader.* Heidelberg: J. Groos.

Guarino, N. & Poli, R. (Ed.). (1993). *Proceedings of the 1st International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation.* Padova, Italy: Ladseb-CNR, Institute for Systems Theory and Biomedical Engineering of the Italian National Research Council.

Harel, D. (1987). *Algorithmics, The Spirit of Computing.* Addison-Wesley.

Hart, A. (1986). *Knowledge Acquisition for Expert Systems.* McGraw-Hill.

Jain, A. K. & Dubes, R. C. (1988). *Algorithms for Clustering Data.* Englewood Cliffs, NJ: Prentice Hall.

Kodratoff, Y. (1988). *Introduction to Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Kodratoff, Y. & Michalski, R. S. (1990). *Machine Learning: An Artificial Intelligence Approach.* Morgan Kaufmann Publishers.

Lebowitz, M. (1986). Concept Learning in a Rich Input Domain: Generalization Based Memory. In R. Michalski,J. Carbonell, & T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach,* (pp. 193-214). Los Altos, CA: Morgan Kaufmann.

Martinez, J. L. J. & Kesner, R. P. (Ed.). (1991). *Learning and Memory, A Biological View.* Academic Press.

Michalski, R., Carbonell, J. & Mitchell, T. (1983). *Machine Learning: An Artificial Intelligence Approach.* Tioga Publishing Company.

Michalski, R., Carbonell, J. & Mitchell, T. (1986). *Machine Learning: An Artificial Intelligence Approach.* Los Altos, CA: Morgan Kaufmann.

Michalski, R. S. & Stepp, R. E. (1983). Learning from Observation: Conceptual Clustering. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach,* (pp. 331-364). Morgan Kaufmann Publishers.

Michalski, R. S., Stepp, R. E. & Diday, E. (1981). A Recent Advance in Data Analysis: Clustering Objects into Classes Characterized by Conjunctive Concepts. In L. N. Kanal & A. Rosenfeld (Eds.), *Progress in Pattern Recognition,* (pp. 33-56). North-Holland Publishing Co.

Mineau, G. (1990) *Structuration des bases de connaissances par généralisation.* Ph.D., Universit de Montral.

Mineau, G., Gecsei, J. & Godin, R. (1990). Structuring Knowledge Bases using Automatic Learning. In *Proceedings of the IEEE Sixth Int'l Conf. on Data Engineering,* Los Angeles, CA: IEEE Computer Society Press, pp. 274-280.

Mineau, G., Godin, R. & Missaoui, R. (1993). Assisting Domain Analysis by Conceptual Clustering for Software Reuse. Submitted for publication.

Mineau, G. W. (1992). Normalizing Conceptual Graphs. In T. Nagle,J. Nagle,L. Gerholz, & P. Eklund (Eds.), *Conceptual Structures: current research and practice,* (pp. 339-348). Ellis Horwood.

Mineau, G. W. (1993a). Facilitating the Creation of a Multiple Index on Graph-Described Documents by Transforming Their Descriptions. In *Proceedings of the 2nd Int. Conf. on Information and Knowledge Management,* Washington, USA: to appear.

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

Mineau, G. W. (1993b). Sharing Ontologies is Sharing Semantics. In *Proceedings of the Formal Ontology in Conceptual Analysis and Knowledge Representation*, Padova, Italy. March.

Mineau, G. W. & Godin, R. (1992). Automatic Knowledge Structuring for Browsing Retrieval. In *Proceedings of the 1st Int. Conf. on Information and Knowledge Management (CIKM-92)*, Y. Yesha (Ed.), Baltimore, USA: Int. Society of Mini and Microcomputers (ISMM), pp. 273-281.

Mineau, G. W. & Godin, R. (1993). Automatic Structuring of Knowledge Bases by Conceptual Clustering. Submitted for publication.

Mineau, G. W., Godin, R. & Missaoui, R. (1993a). Assisting Domain Analysis by Conceptual Clustering for Software Reuse. Submitted for publication.

Mineau, G. W., Godin, R. & Missaoui, R. (1993b). Induction of Generic Data Models by Conceptual Clustering. In *Proceedings of the Fifth Int. Conf on Software Engineering and Knowledge Engineering*, San Francisco, USA: IEEE Press.

Puff, C. R. (Ed.). (1979). *Memory Organization and Structure*. Academic Press.

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.

Sowa, J. F. (Ed.). (1991). *Principles of Semantic Networks: Exploration in the Representation of Knowledge*. Morgan Kaufmann Publishers.

Sowa, J. F. (1993). Representing Attributes, Roles, and Nondetachable Parts. In *Proceedings of the 1st International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation,* N. Guarino & R. Poli (Ed.), Padova, Italy: Ladseb-CNR, Institute for Systems Theory and Biomedical Engineering of the Italian National Research Council.

Stepp, R. E. & Michalski, R. S. (1986). Conceptual Clustering: Inventing Goal-Oriented Classifications of Structured Objects. In R. Michalski, J. Carbonell, & T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach,* (pp. 471-498). Los Altos, CA: Morgan Kaufmann.

ISSN: 2324-9773

Mineau, G. (1993). The Classification of Structured Knowledge Objects. 4th ASIS SIG/CR Classification Research Workshop, 165-178. doi:10.7152/acro.v4i1.12618

ISSN: 2324-9773