Conceptual Knowledge Processing with Formal Concept Analysis and Ontologies

Philipp Cimiano, Andreas Hotho, Gerd Stumme, Julien Tane

Institute for Applied Informatics and Formal Description Methods (AIFB)
University of Karlsruhe, D–76128 Karlsruhe, Germany
www.aifb.uni-karlsruhe.de/WBS, [lastname]@aifb.uni-karlsruhe.de

Abstract. Among many other knowledge representations formalisms, Ontologies and Formal Concept Analysis (FCA) aim at modeling 'concepts'. We discuss how these two formalisms may complement another from an application point of view. In particular, we will see how FCA can be used to support Ontology Engineering, and how ontologies can be exploited in FCA applications. The interplay of FCA and ontologies is studied along the life cycle of an ontology: (i) FCA can support the building of the ontology as a learning technique. (ii) The established ontology can be analyzed and navigated by using techniques of FCA. (iii) Last but not least, the ontology may be used to improve an FCA application.

1 Introduction

As concepts are the most basic units of thought, it is not surprising that they became important building blocks in Artificial Intelligence research. Their appearance is prevailing in Knowledge Representation (e. g., in semantic networks, conceptual graphs, description logics), but they also appear for instance in Machine Learning (e. g., in conceptual clustering, concept learning). All these approaches focus on other aspects of concepts, leading to different formalizations.

In this paper, we focus on two of these formalizations, namely Ontologies and Formal Concept Analysis. We will analyze how the two approaches can complement each other from an application point of view. In particular, we will discuss how Formal Concept Analysis can be used to support Ontology Engineering, and how ontologies can be exploited in FCA applications. We will here illustrate these aspects by selected applications which we have set up during the last years.

The interplay of FCA and ontologies can be seen along the life cycle of an ontology:

1. FCA can support the building of the ontology as a learning technique.
2. The established ontology can be analyzed and navigated by using techniques of FCA.
3. Last but not least, the ontology may be used to improve an FCA application.

The role of FCA in these three steps of the ontology life cycle is discussed in Sections 3 to 5, resp. Each of these sections provides two examples, and concludes with a discussion of related work. The first example in each section has been presented already sometime ago, we recall it only briefly here to sketch the whole picture. The second example in each of the three sections is more recent, and is discussed in more depth. The discussions of related work are rather selective in that we consider only articles which address explicitly the combination of Formal Concept Analysis and ontologies. Before having a look at these applications, we will start in the next section with a more in-depth discussion of the relationship between ontologies and Formal Concept Analysis.

2 Concepts in Ontologies and in Formal Concept Analysis

'Ontology' in its original sense is a philosophical discipline dealing with the potentialities and conditions of being. Within Computer Science, 'ontologies' have been introduced about a decade ago as a means for formally representing knowledge. Following [9], they are considered as an "explicit, [formal,] specification of a [shared] conceptualization [of a domain of interest]".[1] This means that ontologies serve as representation in some pre-defined formalism of those concepts and their relationships which are needed to model a certain application domain. Three major uses of ontologies can be distinguished: communication (between machines and/or humans), automated reasoning, and representation and reuse of knowledge. Even though it seems at first glance that the use of the term 'ontology' differs a lot in philosophy and computer science, W. Hesse pointed out in [11] that its computer science understanding fits rather well with interpretations of last century philosophy.

As we have just seen, ontologies have the ontological status of a model. Their purpose is to model a shared understanding of the reality as perceived by some individuals in order to support knowledge-intensive applications. Formal Concept Analysis, on the other hand, plays a different role. Concept lattices are not understood as modeling some part of the reality, but rather as an artifact, which is derived from some dataset. This artifact is intended to support the user in analyzing and structuring the domain, based on the given data. While ontologies can be established without any given data, FCA relies thus always on some set of objects. Thus, in FCA, extensional and intensional aspects are equally important, while ontologies emphasize on the intensional part.

The interaction between FCA and ontologies may go in two directions. On one hand, FCA can be used as a technique for Ontology Engineering. It supports the structuring of some given data by means of concept lattices. They can be used either to extract, from a given dataset, a conceptual hierarchy which may serve as a basis for the manual or semi-automatic development of an ontology. Or they are used for visualizing the ontology, in order to support navigation and analysis tasks. For both aspects, there are basically two ways how FCA and ontology notions can be combined. The most obvious way from a theoretical viewpoint is to identify the ontology concepts with the formal concepts of FCA. In many applications, however, it turns out that the canonical match is between the ontology concepts and the FCA attributes. While FCA theory forces a distinction between concepts and attributes, that distinction is not that sharp in the ontology world. Or, as the German standard DIN 2330 states, attributes "are units of thought which are gained by abstraction, and hence they are also concepts. For building concepts, one always needs other concepts, which then play the role of

[1] Gruber's original version is without the words 'formal', 'share', and 'of a domain of interest', which nowadays are rather accepted to describe more precisely the intention of ontologies with Computer Science.

attributes" (translated from [?] by the authors). In fact, the decision if something is to be modeled as an attribute or as a concept, is a discussion which is always coming up in Ontology Engineering. It is though an interesting research topic, how the dual role of attributes and concepts can be incorporated in FCA theory. In Sections 3 and 4, we will see how the two ways of matching ontology concepts with formal concepts or with FCA attributes show up in selected applications.

On the other hand, ontologies can be used to improve FCA applications. In standard Formal Concept Analysis, the set of attributes does not carry any structure. By considering this set as a set of ontology concepts, we can model relations and dependencies between the attributes. Although this does not increase the complexity of the resulting lattices (as concept lattices cover, up to isomorphism, the whole class of complete lattices), it enriches the conceptual structure and provides new means of interaction and analysis. In Section 5, we will discuss two applications of this conceptual enrichment.

Beside those two directions of interaction — which may be closed to a loop — there is also the possibility of a tighter integration. One attempt is for instance Wille et al's work on Contextual Logic, where FCA is considered as a theory for concepts as basic units of thought, while concept(ual) graphs are formalizing judgments and conclusions [?,?] (see also [22]). This combination of FCA and conceptual graphs aims thus at one unifying theory for formalizing traditional logic based on the triad concept – judgment – conclusion. This tighter integration, however, will not be discussed here. In the following sections we focus rather on the first two ways of interaction.

3 Ontology Learning with Formal Concept Analysis

As mentioned above, an ontology is an explicit specification of a conceptualization. However, in reality most conceptualizations are not made explicit but are rather implicit in documents, people's heads or even in actions carried out by them. In this sense, a big challenge is to externalize the knowledge implicitly contained in these sources and to crystallize it into an ontology, i.e. a formal and explicit conceptualization.
It is also not realistic to assume that there will be one single and complete ontology for a given domain of interest. In fact, it is common in our world that different views on a certain topic or domain co-exist, each of them with their own level of detail, granularity, completeness and with their own focus. Thus, a very important issue is to combine the view of different parties on a certain domain, i.e. to merge their respective ontologies. In this section we address these two very important topics: In Section 3.1, we tackle the problem of merging existing ontologies together with the help of FCA. In Section 3.2, we focus then on the externalization of the knowledge which is implicit in texts and show how FCA can be exploited for this purpose.

3.1 Ontology Merging

The process of *ontology merging* takes as input two (or more) source ontologies and returns a merged ontology based on the given source ontologies. The resulting ontology may then be used for translating between applications which are based on their respective source ontologies. High quality results of the merging process will always need

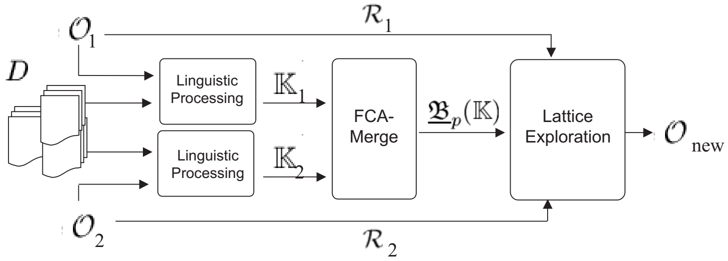

Fig. 1. The FCA–MERGE approach

a human involved who is able to make judgments based on background knowledge, social conventions, and purposes. Thus, all merging approaches aim at supporting the knowledge engineer, and not at replacing him.

In [**?**], we presented the method FCA–MERGE for merging ontologies following a bottom-up approach and offering a global structural description of the merging process. For each source ontology, it extracts instances from a given set of domain-specific text documents by applying natural language processing techniques (see left part of Figure 1). This way, one context is computed for each source ontology. Its objects are the documents, and its attributes are the ontology concepts. An ontology concept is related to a document iff it occurs in the document. The contexts are joined by context apposition, and a pruned concept lattice is computed with the TITANIC algorithm [**?**] (see middle of Figure 1). The concept lattice provides a hierarchical, conceptual clustering of the concepts of the source ontologies. It is explored and interactively transformed to the merged ontology by the ontology engineer.

Observe that, in this approach, the ontology concepts have been identified with the FCA attributes, and not with the formal concepts. This seems to be the natural approach, as they appear already as input to the FCA step, while formal concepts only show up at the end of this step.

3.2 Ontology Learning from Text

There is no doubt that oral communication or text understanding presuppose some sort of *common ground* between the communicating partners or between the writer and the reader of a text [8]. However, this common ground or conceptualization is typically not made explicit in the conversation or the text itself. Brewster et al. [3] for example have argued that text writing and reading is in fact a process of background knowledge maintenance in the sense that basic domain knowledge is assumed, and only the relevant part of knowledge which is the issue of the article is mentioned in a more or less explicit way. Actually, knowledge can be found in texts at different levels of explicitness depending on the sort of text considered. Handbooks, textbooks or dictionaries for example contain explicit knowledge in form of definitions such as “a tiger is a mammal” or “mammals such as tigers, lions or elephants”. In fact, some researchers have exploited such regular patterns to discover taxonomic or meronymic relations in texts [10, 5]. However, it

Table 1. Object/Verb Relationships as formal context

	bookable	rentable	driveable	rideable	joinable
hotel	x				
apartment	x	x			
car	x	x	x		
bike	x	x	x	x	
excursion	x				x
trip	x				x

seems that the more technical and specialized the texts get, the less basic knowledge we will find in them stated in an explicit way. Thus, an interesting alternative is to derive knowledge from texts by analyzing how certain terms are used rather than to look for explicit definitions of them. In this line the *distributional hypothesis* assumes that terms are similar to the extent to which they share similar linguistic contexts. Verbs for example give information about the state in which certain objects are or about which actions are carried out on them.

Let's for example assume that we are interested in deriving some sort of conceptualization for the tourism domain by analyzing texts related to this area. By looking at verbs as well as their direct objects we could for example derive a formal context as depicted in Table 1. If we now make the underlying assumption that the relations in Table 1 are more or less complete in the sense that we have a 'closed world' and all non-occurring relations in the text are regarded as negative examples, we could group the objects into classes or even into a concept hierarchy by analyzing their shared linguistic contexts. This is in fact the assumption of most clustering techniques which have tried to group terms appearing in texts into meaningful classes or even term hierarchies [12, 1, 15, 4, 20, 2]. In this line it is an interesting option to make use of FCA to structure such terms into abstract units or concepts.

The concept lattice of the formal context in Table 1 is depicted in Figure 2. It can be transformed into a partial order as shown in the same figure in a straightforward way by removing the bottom element, introducing an ontological concept for each formal concept (named with the intent) and introducing a subconcept for each element in the contingent of the formal concept in question. In fact, this is the aim of our ongoing work presented in [6] and [7]. In order to derive the necessary verb/object dependencies from text we make use of a natural language parser, which produces a syntactic tree for each sentence in the text and from which these dependencies can be obtained in a straightforward way. Furthermore, we normalize the extracted verbs and objects by lemmatizing them via a lexicon lookup, i.e. *bought*/*buys* are transformed to the infinitive *buy*, and *hotels* is transformed to the singular form *hotel*. Furthermore, we also add the postfix '-able' to verbs to make them look more like attributes and to facilitate human understanding of our automatically derived lattices and concept hierarchies. In this context, there are three important issues to consider:

1. the output of the parser can be erroneous, i.e. not all derived verb/object dependencies are correct,

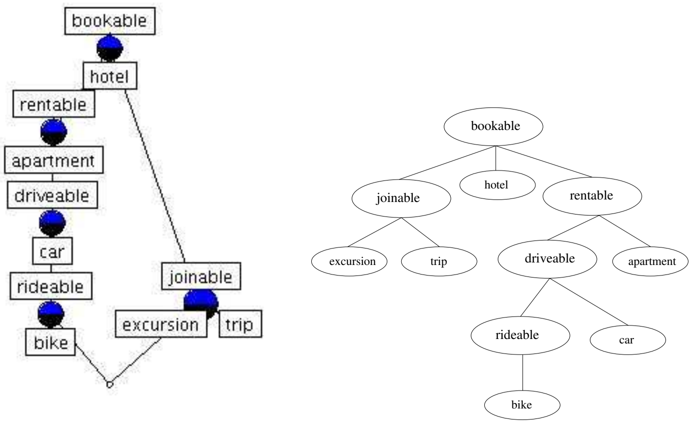

Fig. 2. The lattice and partial order for the tourism example

2. not all the derived dependencies are 'interesting' in the sense that they will help to discriminate between the different objects,
3. the assumption of completeness of information will never be fulfilled, i.e. the text collection will never be big enough to find all the possible occurrences (compare [23]).

To deal with the first two problems, we weight the verb/object dependencies with regard to a certain information measure. In [6] we explored different measures and found out that the simple conditional probability works well enough for our purposes. The use of such an information measure deals with the first two problems as we consider only those verb/object relations for which the mentioned information measure is above some threshold t. The third problem is dealt with by our 'Smooth FCA' approach presented in [7]. In this approach we iteratively cluster terms and verbs together which are mutually most similar with respect to a certain similarity measure and thus artificially create more attribute/object pairs. The result is an overall 'smoothing' of the relative frequency landscape by assigning some non-zero relative frequencies to combinations of verbs and objects which were actually not found in the corpus. Our experiments so far have actually shown that the use of this smoothing technique actually improves the quality of the learned concept hierarchies [7]. Figure 3 shows a lattice which was automatically derived from a set of texts acquired from *http://www.lonelyplanet.com* as well as *http://www.all-in-all.de*, a webpage containing information about the history, accommodation facilities as well as activities of *Mecklenburg Vorpommern*, a region in northeast Germany. We only extracted verb/object pairs for the terms in Table 1. The corpus size was about a million words and we used the *Resnik* measure described in [6] to weight the extracted verb/object pairs. As it can be concluded from Figure 3, for *excursion* no verb/object dependencies were found in the corpus.

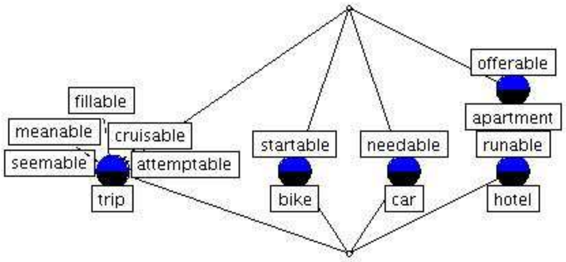

Fig. 3. The lattice automatically derived from tourism-related texts

3.3 Related Work

The idea of using FCA in NLP in general is certainly not new. In [19] for example, several possible applications of FCA in analyzing linguistic structures, lexical semantics and lexical tuning are mentioned. [21] and [16] apply FCA to yield more concise lexical inheritance hierarchies with regard to morphological features such as numerus, gender etc. However, to our knowledge, FCA has not been applied before to the acquisition of domain concept hierarchies such as in the approach described in this paper.

FCA–Merge is not the first tool for supporting the merging process of ontologies. The need for some support in the merging process showed up soon, since manual ontology merging using conventional editing tools is rather difficult, labor intensive and error prone. Therefore, several systems and frameworks have recently been proposed [**?**,**?**,**?**,**?**]. They rely on syntactic and semantic matching heuristics which are derived from the behavior of ontology engineers when confronted with the task of merging ontologies, i. e., human behavior is simulated. Although some of them locally use different kinds of logics for comparisons, these approaches do not offer a structural description of the global merging process.

Up to our knowledge, there is no other publication on FCA-based ontology merging, with one exception, namely OntEx, which we presented in [**?**]. OntEx is based on Bernhard Ganter's Attribute Exploration. Its starts with the concepts of the source ontologies, and with all known hierarchical relationships. It interacts with the knowledge engineer by asking him, if further hierarchical relationships hold. If the user agrees, then the relation between the concepts is added, if he denies, then he has to provide a counterexample, which is exploited in the sequel. As FCA–Merge, OntEx also gives some guarantee that all possible combinations are considered. However, to be applicable in practice, it has to be combined with heuristic methods, as there is a price to pay for this guarantee because of the high number of interactions.

4 Navigating ontologies with FCA

Ontologies are most often too large to be displayed on a single computer screen. Therefore, visualization and interaction techniques are needed, if the intended use of the ontology comprises a direct user access to the ontology and/or knowledge base. This is for instance the case for ontology-based web portals (see, e. g., the OntoWeb portal[2]). In this section, we present two applications where Formal Concept Analysis supports the navigation within an ontology-based application: the first system is an email management system, the second a management system for distributed learning repositories.

4.1 Conceptual Email Manager

The way standard email management systems store mails is directly derived from the tree structure of filing cabinets and file management systems. This has the advantage that trees have a simple structure which can easily be explained to novice users. The disadvantage is that at the moment of storing an email the user already has to foresee the way she is going to retrieve the mail later. The tree structure forces her to decide at that moment which criteria to consider as primary and which as secondary. This problem arises especially if a user communicates with overlapping communities on different topics.

In [**?**] (see also [**?**]), we presented the *Conceptual Email Manager CEM*.[3] It uses a simple ontology for storing the emails. The ontology consists of a hierarchy of concepts (catchwords) together with a lexicon. The hierarchy of the ontology may be any partially ordered set; multiple inheritance is thus explicitly allowed. Emails may be assigned to more than one concept, i. e., multiple instantiation is also allowed. In fact, the advantages of FCA support for navigation and retrieval tasks only comes to its full potential, when multiple inheritance and/or multiple instantiation show up.

From a FCA perspective, multiple instantiation can be seen as storing the emails in a formal context. The objects are the emails, and the attributes are the catchwords. In this setting, FCA attributes and ontology concepts are thus considered as equivalent. (Multiple) inheritance is then modeled as an additional partial order on the set of attributes, which has to be reflected by the binary relation of the context: if an email is related to a catchword, then it also has to be related to all catchwords which are higher in the partial order. This allows to assign explicitly only very specific attributes to an email (e. g,. 'ICFCA 2004'), while more general attributes are then inherited (e. g., 'Conference submission'). Different terms in different languages can be assigned to the same attribute within the lexicon. It allows to extract the assignment of attributes to emails automatically by means of Natural Language Processing techniques, reducing thus the workload of the user.[4]

Multiple inheritance and multiple instantiation allow the user to retrieve emails via a concept lattice following different paths. This means that one does not need to decide

[2] http://ontoweb.aifb.uni-karlsruhe.de/

[3] A commercial follow-up of the Conceptual Email Manager is offered by Email Analysis Pty Ltd, Sydney, see http://www.mail-sleuth.com .

[4] If wanted, the user can still modify the result afterwards.

which of the paths to use for storing. For retrieving the mail later, one can consider any combination of the catchwords. This is made possible by using the richly structured concept lattice as search space.

The ontology provides also a mechanism for automatically generating new conceptual scales. Each catchword gives rise to one conceptual scale, which has as attributes all its immediate subconcepts. The set of scales inherits the hierarchy from the ontology, so that one can focus into a single concept by selecting a next, more specific scale. This kind of 'conceptual focusing' is visualized by means of local scaling [**?**].

4.2 Courseware Watchdog

On a personal computer, it is possible to organize resources according to personal needs. In the case of remotely stored resources, this is not possible anymore, since their storage is not under the control of the user. Through the use of hypertext, remote material can be linked and retrieved when needed, but the particular problem of finding and organizing this remote material becomes even more crucial.

Research in the E-Learning domain shows that standards are needed for interoperability, but true interoperability does not only need data integration, it also has to consider the integration of applications. In [**?**], we illustrate such an integration of E-learning related applications with the implementation of a Courseware Watchdog. It is part of the PADLR project (Personalized Access to Distributed Learning Repositories) that builds upon a peer-to-peer approach for supporting personalized access to learning material.[5]

The Courseware Watchdog is built on top of the Karlsruhe Ontology and Semantic Web Framework KAON [**?**].[6] KAON offers abstractions for ontologies and text corpora, an ontology editor and application framework, inferencing, and persistence mechanisms, etc. The Watchdog consists of the following components which are organized around an ontology management system:

1. *Visualization and interactive browsing techniques* allow the browsing of ontology and knowledge base in order to improve the interaction between of the user with the content.
2. A *focused crawler* finds related web sites and documents that match the user's interests. The crawl can be focused by checking new documents against the user's preferences as specified in terms of the ontology.
3. An *Edutella peer* enables querying for meta-data on learning objects with an expressive query language, and allows to publish local resources in the P2P network.
4. A *subjective clustering component* is used to generate subjective views onto the documents.
5. An *ontology evolution component* comprises ontology learning methods which discover changes and trends within the field of interest.

Here, we focus on the first component.

[5] http://www.learninglab.de/english/projects/padlr.html
[6] http://kaon.semanticweb.org

Ontologies are based two kinds of relations: hierarchical and non-hierarchical relations. For each kind of relation, we use an appropriate technique: the display of hierarchies through concept lattices in the first case, and relational browsing else. The browsing component is implemented in the KAON framework, and extends the Concept Explorer[7] as a library.

We display the subsumption hierarchies of the ontologies (and in a similar way other hierarchical relations as for instance property or topic hierarchies) by embedding them into their Dedekind-McNeille completion, i. e., by displaying the concept lattice of the formal context $(C, C, \leq)$, where C is the set of concepts of the ontology, and $\leq_C$ is the subsumption hierarchy. Unlike tree-oriented visualization techniques, this allows for the display of multiple inheritance. As in the Conceptual Email Manager, we also introduce instances and allow multiple instantiation. This is displayed as the concept lattice of the formal context $(C \cup I, C, \leq_C \cup \text{ is-a})$ where I is the set of instances, and is-a is the instantiation relation between I and C. Using this approach, some new concepts may appear, even if they were not explicitly modeled within the ontology. These concepts may be used as a hint to the user to name them and to include them in his ontology.

In contrast to the Conceptual Email Manager, the Courseware Watchdog also has to consider non-hierarchical relations in the ontology. These relations represent links between diverse elements of the ontology (e. g., the "lecturer" of a "course" should be linked to it by a relation "holdsCourse"). These kinds of relation are best understood and used through some kind of exploration. Relational browsing is a technique consisting in offering the user different links which he can choose to follow. In addition to normal browsing along hyperlinks, the links are typed according to the ontology. It is possible to navigate and explore the ontology following the relations of the ontology, and then display different kinds of hierarchies.

The browsing component is also a generic way of interacting with the ontology within the other modules of the Courseware Watchdog. I. e., the user can select concepts in the Hasse diagram for use in the crawling, clustering or evolution process or for querying the P2P network.

Within the framework of Formal Concept Analysis, ontologies can be considered (under some constraints) as (many-valued) multi-contexts. We are currently investigating how to formalize this relationship, and how to exploit it for a tighter integration. In particular, we want to enhance further the interaction between FCA navigation metaphors and ontology-based knowledge management techniques.

4.3 Related Work

Systems like WAVE for web navigation [**?**], the expert system MCRDR [**?**], the RFCA system for browsing rental advertisements on the WWW [**?**] were first prototypes integrating both FCA and ontologies. The work on the Conceptual Email Manager was inspired by these systems. Is is also related to the use of *virtual folders* in the program View Mail (VM) [**?**], which are collections of email documents retrieved in response to a query.

[7] http://www.sourceforge.net/projects/conexp

A promising next step along the line of the Conceptual Email Manager and the Courseware Watchdog will be to establish interfaces between the KAON framework and the research and software project 'Tockit — Framework for Conceptual Knowledge Processing'[8], in order to obtain a large, stable platform for future FCA/ontology-based projects.

5 Using Ontologies for Formal Concept Analysis Applications

One of the criteria that made Formal Concept Analysis successful is certainly the fact that, for most applications, only rather simple mathematical concepts are needed — the simplest just being a binary relation between two sets. Surprisingly many applications benefited from FCA just in this basic form. However, there were also many applications which showed the need of a more expressive knowledge representation. In this line arose for instance many-valued contexts, multicontexts, triadic contexts, and the like. The challenge for research in Formal Concept Analysis was (and still is) to bring together these more complex kinds of knowledge representation with the charm of the natural appearance of a (concept) lattice out of the Galois connection induced by a binary relation. In the next subsection, we will discuss one way how to derive a concept lattice out of knowledge represented in a description logic.

Another important problem in FCA applications is the size of the concept lattices. In the worst case, their size is exponential in the size of the formal context. Hence methods for managing and structuring large conceptual hierarchies and for visualizing (parts of) them have been developed. These include for instance conceptual scaling, local scaling, and iceberg concept lattices. In the second part of this section, we show a new, orthogonal approach: By summarizing 'similar' objects prior to the lattice computation with a standard clustering algorithm together with background knowledge encoded in an ontology, we are able to reduce significantly the size of the concept lattice without loosing too much information.

5.1 Defining FCA attributes with Description Logics

The best known approach for deriving a concept lattice from a dataset which is more than just one binary relation is conceptual scaling. It allows to derive unary attributes from many-valued ones, which are then the input to the lattice computation. However, conceptual scaling still requires that the data is represented in one (database) relation with the object name being a primary key. Conceptual scaling is thus not able to deal with more than one relation. In FCA, the problem of multiple relations has been encoded in the definition of (many-valued) multicontexts [**?**,**?**], but a theory analog to conceptual scaling, which allows the transformation of a multicontext into a meaningful structure of concept lattices has not completely worked out up to now.

However, first steps in this direction have been made. One of them is logical scaling. In [17], S. Prediger showed how a formal context (and thus a concept lattice) can be derived from a database, once a set of new attributes together with definitions in

[8] http://tockit.sourceforge.net/

drinks	
Person	Wine
Mr. Smith	Casa Solar
Mrs. Miller	Staehle
Miss Cooper	Figeac
Mr. Davis	Figeac
Mr. Davis	Casa Solar

Wines				
	red wine	white wine	Bordeaux	Price
Figeac	×		×	49,90
Staehle		×		14,90
Casa Solar	×			5,95

wine drinker	$:=$ person $\sqcap$ $\exists$ drinks.(red wine $\sqcup$ white wine)
red wine drinker	$:=$ person $\sqcap$ $\exists$ drinks.red wine
white wine drinker	$:=$ person $\sqcap$ $\exists$ drinks.white wine
Bordeaux drinker	$:=$ person $\sqcap$ $\exists$ drinks.Bordeaux
red Bordeaux drinker	$:=$ person $\sqcap$ $\exists$ drinks.(red wine $\sqcap$ Bordeaux)

Fig. 4. A many-valued multi-context (top) and the definition of a logical scale (bottom).

some logical language is given. In [18], we extended this approach to pre-definable conceptual scales. Logical scales can thus seamlessly be used together with 'standard' conceptual scales. In particular, they can be combined in nested line diagrams.

In order to illustrate the approach, we discuss a small example (see also [18]): Consider the database consisting of the two relations shown in the upper half of Figure 4. In terms of FCA it is a many-valued multicontext, in terms of Description Logics (DL) it is an A-Box. Assume that we want to classify the persons according to their wine drinking behavior. With the set of definitions shown in the lower part of Figure 4 (T-Box in DL terminology), we can define typical profiles of wine fans.[9] These definitions yield two possible scales: the data-driven and the theory-driven logical scale.

The realized data-driven scale [17] of our example is shown in the left of Figure 5. It is derived from the database by restricting the set of instances to those of the concept 'Person', and by taking all defined concepts of the T-Box as attributes. From the diagram, one can for instance read that all Bordeaux drinkers are also drinking red wine. However, it is not clear from the diagram if this implication holds for all possible objects (i. e., if it is forced by the definitions in the T-Box) or if it is just a coincidence because our set of instances does not cover all proto-typical cases.

Theory-driven scales [18] take this distinction into account. They consider all possible conjunctions of the defined attributes in the T-Box. The scale is computed with B. Ganters Attribute Exploration using a DL subsumption algorithm as 'expert' (see [**?**] and Baaders article in this volume). The realized theory-driven scale of our example is shown in the right side of Figure 5. The data-driven scale is embedded in it as a join-semi-lattice. In the diagram, we can read the implications based on the actual set of instances as well, but it also shows which further attribute combinations are possible. For instance, one can see that the definitions do not force Bordeaux drinkers to drink red

[9] For a description of how to read these definitions, refer to the article of F. Baader in this volume.

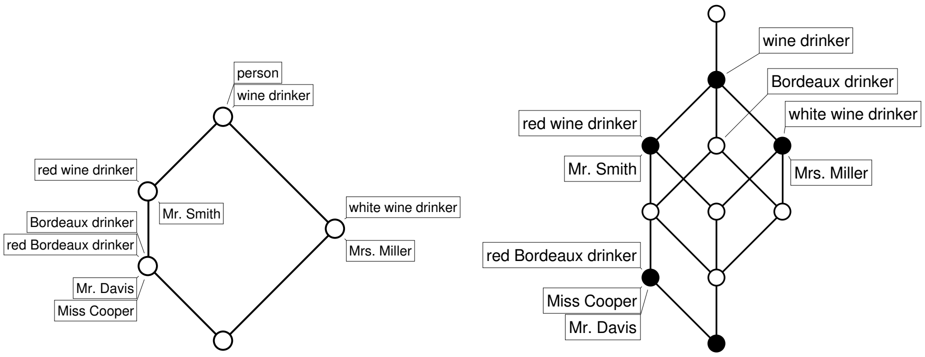

Fig. 5. The resulting data-driven (left) and theory-driven realized scales.

wine (they may also drink white Bordeaux only). One can also see that 'wine drinker' is the most general of the defined attributes, but that there may also exist instances not belonging to it, as this concept is different from the top concept.

The approaches presented here can be applied to any DL, provided that the instance problem is decidable within that language. I. e., one needs a sound and complete algorithm which determines whether an instance from the A-Box belongs, in each model, to a given concept from the T-Box or not. If there are more than five to six definitions within the T-Box, they have to be grouped thematically into smaller subsets in order to become scales of reasonable size; each subset is giving rise to one logical scale.

5.2 Preprocessing large datasets for FCA

For a long time, conceptual scaling has been considered as *the* technique for dealing with large datasets in FCA: if one projects the data set only on those attributes the user is currently interested in, then the size of the lattice remains small enough to be displayed on a computer screen or on a sheet of paper. Conceptual scaling is still a very useful technique when the user has an idea which attributes (or which conceptual scales) are relevant to his task at hand. However, if the user is first confronted with a dataset, he needs some support to gain first insights, before he can start to select systematically attributes or conceptual scales.

One solution to this problem are iceberg concept lattices [**?**]. They can be computed without requiring any initial effort from the user, and provide a view on the top level of the concept lattice. Starting from this, the user can then continue the exploration with conventional means such as conceptual scaling (e. g., with TOSCANA [**?**]) or by locally analyzing the concept lattice (e. g., with the GALOIS system [**?**]).

Here, we will discuss another approach: If two objects are almost but not completely identical (according to the context), then they give rise to different object concepts, no matter how small the difference is. This effect can be propagated all along the concept lattice. Hence, if we allow to identify objects which are 'almost identical', then we may significantly reduce the size of the concept lattice, without loosing too many details. The resulting — relatively small — lattice can be used as a starting point for an exploration. If one wants to analyze the data on a more detailed level, where also minor differences between objects have to be considered, one may still come back to the original concept lattice (and analyze it using conceptual scaling or local focusing [**?**]).

We tested this approach in a text mining scenario [**?**,13], where we studied the use of background knowledge in form of an ontology for clustering sets of text documents. We selected the Reuters-21578[10] text collection for our experiments. The corpus consists of 21578 documents, and is especially interesting for the evaluation of clustering algorithms, as it comes along with a hand-crafted classification.

We performed the usual preprocessing steps for text mining within TextToOnto,[11] a text mining system we developed at our institute within the KAON framework: extracting a bag-of-words model (vector space model), stopword removal, stemming, and tfidf weighting. After that, each document is represented as a vector in the vector space $\mathbb{R}^n$ where n is the number of remaining word stems. The ith component of the vector of a document indicates the weighted frequency of the ith word stem within the document.

The vector space can also be seen as a many-valued context: each document is an object, each word stem an attribute, and the attribute values are the weighted frequencies. By conceptual scaling (e. g., by nominal scaling), one can transform this context into a one-valued one, and can compute the concept lattice. We performed this for the Reuters dataset, but were not convinced from the results: it was very difficult to find the right values for the nominal scales. Hence the lattice was either much too detailed to provide any insight, or we obtained a flat hierarchy.

Our next step was to cluster the documents using the BiSec–k–Means algorithm before applying FCA. Instead of the individual documents, we consider now the clusters as objects of the formal context. The set of attributes remains unchanged, and the attribute values where taken from the centroid vector of each cluster. This clustering step is a generalization of the FCA task of purifying the object set of the context, in that it does not only consider completely identical objects (in the sense of the context) to be identified, but also 'almost identical' ones. Especially in text mining this makes sense, as almost no two documents have exactly the same representation in the bag-of-words model.

While this pre-clustering provided concept lattices that were smaller than the ones obtained by conceptual scaling, and were thus easier to interpret, they still suffered from the problem that they were rather flat. The problem is that documents may be closely related even though they use different expressions, which cannot be identified by pure syntactical methods. For instance, one document may talk about 'beef' and another one about 'pork', but beef and pork will be considered as unrelated dimensions in the vector space. This is now where background knowledge in form of an ontology comes into play: if we know that both 'beef' and 'pork' are subsumed under 'meat', then the cluster algorithm can assign both documents to the same cluster.

Before clustering, we extended thus our vector space with concepts of a suitable ontology. In the Reuters scenario, we made use of WordNet,[12] as its coverage fits to the

---

[10] http://www.daviddlewis.com/resources/testcollections/ reuters21578/

[11] http://sourceforge.net/projects/texttoonto/

[12] http://www.cogsci.princeton.edu/~wn/

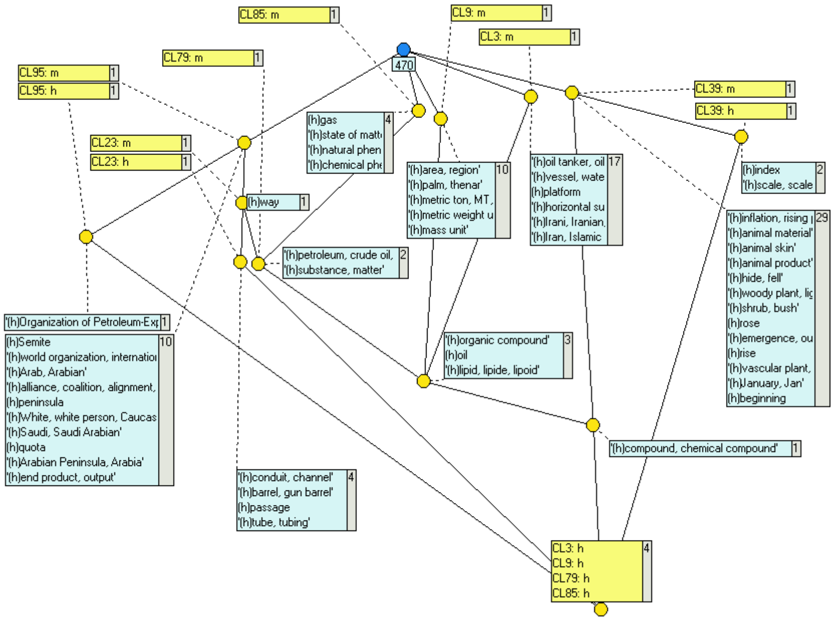

Fig. 6. The resulting conceptual clustering of the text clusters (visualized for the clusters related to chemical compounds.

generality of the Reuters news.[13] Then we applied nominal scaling and computed the concept lattice using the Cernato software of NaviCon Gmbh.[14]

Figure 6 shows an example, where we scaled nominally with two thresholds: 7 % of the maximal value for (at least) *m*edium importance, and 20 % of the maximal value for *h*igh importance. The diagram shows all clusters where the value of the concept (called synset in WordNet) 'compound, chemical compound' in the centroid is above the threshold of 7 %. Due to technical reasons, we reverse the usual reading order: A formal concept $\mathfrak{c}_1$ is a subconcept of a formal concept $\mathfrak{c}_2$ if and only if there is a path of descending(!) edges from the node representing $\mathfrak{c}_1$ to the node representing $\mathfrak{c}_2$.

As an example, consider the concept in the lower middle of the diagram labeled by 'oil'. It has {CL 3 (m), CL 9 (m), CL 23 (m), CL 79 (m), CL 85 (m), CL 95 (m)} as extent, and {organic compound, oil, 'lipid, lipide, lipoid', 'compound, chemical compound'} as intent. Its intent indicates that the majority of documents within these clusters are about oil. This concept has three direct subconcepts: the first has {CL 3 (m)} as extent, and the attributes from above plus some attributes like 'oil tanker' and 'Ira-

[13] In [14], we discuss how different options for resolving ambiguities and for assigning more general concepts influence the quality of the clustering step.

[14] http://www.navicon.de

nian' as intent. The second has $\{\text{CL } 9\,(\text{m})\}$ as extent, and the attributes from above plus some attributes like 'area', 'palm', and 'metric ton' as intent. The third subconcept has $\{\text{CL } 23\,(\text{m}), \text{CL } 79\,(\text{m}), \text{CL } 85\,(\text{m}), \text{CL } 95\,(\text{m})\}$ as extent, and the attributes from above plus 'substance, matter' as intent. These three subconcepts show what distinguishes the clusters related to oil. The majority of documents in Cluster 3 are about transport of oil (from Iran), those in Cluster 9 about (packaging of) palm oil, and those in the remaining clusters about crude oil.

Clustering the objects before applying FCA is an abstraction that might be considered a loss of information. However, it is beneficial for the following reasons. Firstly, it reduces the number of objects such that FCA becomes more efficient. Secondly, the technique is robust with regard to upcoming documents: A new document is first assigned to the cluster with the closest centroid, and then finds its place within the concept lattice. If on the contrary the document is considered directly for updating the concept lattice, there is no guarantee that the structure of the lattice does not change. Finally and most importantly, formal concept analysis applied directly on all documents suffers from the low co-occurrence of terms. The application of FCA on the Reuters-21578 dataset has shown that hardly any two texts generate the same formal concept. Thus, the lattice became large, unwieldy and hard to understand for the human user. Therefore, in our approach we first cluster a large number of texts (e.g. 10^5) into a more manageable number of clusters (e.g. 10^2). Only then we compute the lattice, allowing for abstraction from some randomness in the joint occurrences of terms. In this process, the use of an ontology as background knowledge assures consistency of the results.

5.3 Related Work

Logic has first been applied to extend and structure the set of attributes of a formal context in [**?**]. However, this approach considered all formulas of propositional logic as attributes, which leads to a combinatorial explosion of the concept lattice. Chaudron/Maille [**?**] and Ferré/Ridoux [**?**] presented an approach based on first order logic. These approaches, however, consider all formulas of the given logic as attributes, which leads to an exponential explosion of the concept lattice. Using a T-Box allows to focus on the relevant formulas and to fine-tune the size of the resulting lattice. The way of computing the theory-driven logical scales is following the approach of Baader in [**?**]. In [**?**] is described how this computation can be extended from computing all conjunctions of attributes to computing both conjunctions and disjunctions.

In the future, FCA systems will be able to work on richer knowledge representations, including relational databases, DL, and conceptual graphs. First steps in this direction are already done with various extensions of the management system TOSCANA for Conceptual Information Systems [**?**]; consider, e. g., [**?**,**?**,**?**,**?**,**?**,**?**]. Within text mining, FCA techniques are also expected to be combined with successful techniques of that domain, such as latent semantic indexing [**?**] or concept indexing [**?**].

6 Conclusion

In this paper, we have shown by means of applications that Ontology Engineering and Formal Concept Analysis complement well; and that first steps have been made to set up links between the underlying theories. These links have to be strengthened and are to be exploited for establishing a comprehensive Conceptual Knowledge Processing environment. For future research within Formal Concept Analysis, this means that one has to extend the scope from strongly structured to semi–structured and even unstructured data, allowing to tackle more complex tasks as, e. g., in the Semantic Web. In this line it has become clear that when turning into the analysis of unstructured data and in particular textual data, a major challenge is to process the data in order to reduce their inherent complexity — e. g., by clustering — before actually applying FCA. By this we can reduce the processing time as well as yield more concise and human readable concept lattices also from unstructured data.

Furthermore, it has also become clear that the relation between formal concepts and ontology concepts is far from being trivial and intuitive. In some applications, FCA attributes are interpreted as ontology concepts (cf. Section 4.1), in others formal concepts are transformed into ontology concepts labeled by the intent of the former (cf. Section 3.2). As already mentioned, a major issue is thus to clarify this relation also from a theoretical point of view.

Having a closer look at the ontologies which were involved in our applications, we observe that most of them consist in fact only of a concept hierarchy. Relations have been used only in the Courseware Watchdog (Section 4.2) and in the logical scaling approach (Section 5.1); axioms have only been used in the latter. It is thus an interesting research question, in how far these more expressive knowledge representation features can be exploited further within Formal Concept Analysis.

Concluding, there is a high potential of interesting research questions about establishing and strengthening the link between the related research areas FCA and ontologies — both in theory and applications.

References

1. R. Basili, M.T. Pazienza, and P. Velardi. Hierarchical clustering of verbs. In *Proceedings of the Workshop on Acquisition of Lexical Knowledge from Text*, 1993.
2. G. Bisson, C. Nedellec, and L. Canamero. Designing clustering methods for ontology building - The Mo'K workbench. In *Proceedings of the ECAI Ontology Learning Workshop*, 2000.
3. C. Brewster, F. Ciravegna, and Y. Wilks. Background and foreground knowledge in dynamic ontology construction. In *Proceedings of the SIGIR Semantic Web Workshop*, 2003.
4. S.A. Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguists*, pages 120–126, 1999.
5. E. Charniak and M. Berland. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the ACL*, pages 57–64, 1999.
6. P. Cimiano, S.Staab, and J. Tane. Automatic acquisition of taxonomies from text: FCA meets NLP. In *Proceedings of the International Workshop on Adaptive Text Extraction and Mining*, 2003.
7. P. Cimiano, S. Staab, and J. Tane. Deriving concept hierarchies from text by smooth formal concept analysis. In *Proceedings of the GI Workshop "Lehren Lernen - Wissen - Adaptivität" (LLWA)*, 2003.

8. H. H. Clark. *Arenas of Language Use*, chapter Common Ground and Language Use, Definite Reference and Mutual Knowledge. CSLI, 1992.
9. T.R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In *Formal Analysis in Conceptual Analysis and Knowledge Representation*. Kluwer, 1993.
10. M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, 1992.
11. W. Hesse. Ontologie(n) - Aktuelles Schlagwort. *Informatik Spektrum*, 25(6):477–480, 2002.
12. D. Hindle. Noun classification from predicate-argument structures. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 268–275, 1990.
13. A. Hotho. *Clustern mit Hintergrundwissen*. PhD thesis, Institute AIFB, University of Karlsruhe, 2004. In preparation.
14. A. Hotho, S. Staab, and G. Stumme. Text clustering based on background knowledge. Technical report, University of Karlsruhe, Institute AIFB, 2003. 36 pages.
15. F. Pereira, N. Tishby, and L. Lee. Distributional clustering of english words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190, 1993.
16. Wiebke Petersen. A set-theoretical approach for the induction of inheritance hierarchies. *Electronic NOtes in Theoretical Computer Science*, 51, 2002.
17. S. Prediger. Logical scaling in formal concept analysis. In D.Lukose, H.Delugach, M.Keeler, L.Searle, and J.F. Sowa, editors, *Conceptual structures: Fulfilling Peirce's dream*, Heidelberg. Springer.
18. S. Prediger and G. Stumme. Theory-driven logical scaling. conceptual information systems meet description logics. In E. Franconi et al, editor, *Proc. 6th Intl. Workshop Knowledge Representation Meets Databases*, Heidelberg. CEUR Workshop Proc.
19. Uta Priss. Linguistic applications of formal concept analysis. presentation at the First International Conference on Formal Concept Analysis, Darmstadt, 2003.
20. S. Schulte im Walde. Clustering verbs semantically according to their alternation behaviour. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-00)*, 2000.
21. Caroline Sporleder. A galois lattice based approach to lexical inheritance hierarchy learning. In *Proceedings of the ECAI 2002 Workshop on Machine Learning and Natural Language Processing for Ontology Engineering (OLT 2002)*, 2002.
22. R. Wille. Preconcept algebras and generalized double boolean algebras. In *This volume*.
23. G. Zipf. *Selective Studies and the Principle of Relative Frequency in Language*. Cambridge, 1932.