FCA-Based Ontology Learning From Unstructured Textual Data

S.Jabbari[1,2], K.Stoffel[1]
1. Information Management Institute, University of Neuchatel, Neuchatel 2000, Switzerland
2. Diagnostics Data Science Lab, F. Hoffmann La Roche Ltd. Basel 4070, Switzerland
{simin.jabbari,kilian.stoffel}@unine.ch

***Abstract* — Ontologies have been frequently used for representing a domain knowledge. It has a lot of applications in semantic knowledge extraction. However, learning ontologies especially from unstructured data is a difficult yet an interesting challenge. In this paper, we introduce a pipeline for learning ontology from a text corpora in a semi-automated fashion using Natural Language Processing (NLP) and Formal Concept Analysis (FCA). We apply our proposed method on a small given corpus that consists of some news documents in IT and pharmaceutical domain. We then discuss the potential applications of the proposed model and ideas on how to improve it even further.**

***Index Terms—About* Formal Concept Analysis, Natural Language Processing , Ontology learning, Semantic knowledge extraction.**

I. Introduction

Advancing knowledge in a domain in which one is not familiar with is an interesting yet a difficult challenge. Ontologies have been used as promising tools for knowledge representation by giving structure to unstructured data (such as text). One idea for giving structure to texts is to define a set of concepts and identify relationship between them. In this paper, we focus on how to build an ontology from a text corpus (a bunch of documents that are available in a domain of interest) using Natural Language Processing (NLP) [1], [2] and Formal Concept Analysis (FCA) [3], [4]. NLP and FCA have been recognized as powerful tools for such purposes [5], [6]. The built ontology can be later used for reasoning and semantic knowledge extraction. We mainly focus on the ontology learning and we discuss how one can use such ontology for reasoning and extracting semantic knowledge. As mentioned earlier, the proposed pipeline for such automatic ontology learning is based on applying a sequence of NLP operators to text corpus for text processing, and then a sequence of FCA operators to build taxonomies and concept hierarchies. The latter is then followed by defining relationship between concepts and bringing knowledge expertise to refine the ontology we are trying to learn. In the following, we first describe the sequence of NLP and FCA operators being used for our proposed ontology learning method. We then apply our method to a text corpus which consists of documents telling news about companies in IT and pharmaceutical industries. We further discuss about the potential applications of ontology for indexing and knowledge extraction.

Manuscript received October 9, 2001; revised . (Write the date on which you submitted your paper for review.) This work was supported in part by the U.S. Department of Commerce under Grant BS123456 (sponsor and financial support acknowledgment goes here). Paper titles should be written in uppercase and lowercase letters, not all uppercase. Avoid writing long formulas with subscripts in the title; short formulas that identify the elements are fine (e.g., "Nd–Fe–B"). Do not write "(Invited)" in the title. Full names of authors are preferred in the author field, but are not required. Put a space between authors' initials.

F. A. Author is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (e-mail: author@ boulder.nist.gov).

S. B. Author was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar. colostate.edu).

T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba, Japan (e-mail: author@nrim.go.jp).

II. Methods

In this section, we focus on describing the pipeline for learning ontology from unstructured textual data. Our proposed method consists of three major components (see Fig. 1). In the first component, a formal context is generated as an outcome of text processing and extracting triplets of the form (subject, predicate, object). The second component then generates a set of formal concepts using FCA and builds a concept hierarchy as a taxonomy for our ontology. The third component is then used for converting concept lattice into ontology by defining ontological concepts (from formal concepts) and relationship between them (thanks to the triplets that have been extracted in the first component of the pipeline). The constructed ontology is then improved by incorporating domain knowledge and can be used for generating T-Box (terminological components and general form of relation between different them) from A-Box (i.e., real instances of the T-Box). In the following, we will describe each component with more detail.

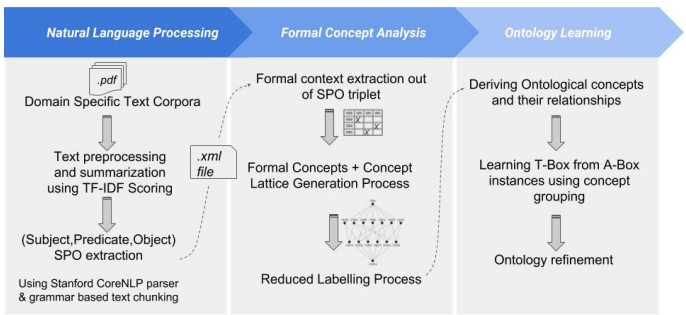

Fig. 1 The pipeline for learning ontology from text corpus consists of three main components. The first step is the preprocessing of the text corpus using NLP. Then a concept hierarchy is generated by FCA. Finally, the concept lattice is transformed into an ontology by defining ontological concepts and the relationship between them, i.e., generating T-Box from A-Box.

A. *Extracting Meaningful Triplets from Text Corpus Using Natural Language Processing (NLP)*

Natural Language Processing (NLP) consists of a set of processing techniques to be applied on textual data for the purpose of text mining. To derive ontology from textual data, the first step is to convert the documents from *.pdf* to *.txt* format, and store them in a text corpus. The next step is then preprocessing of the corpus by *parsing* (i.e., separation of sentences), *tokenization* (i.e., separating words of each sentence), and *POS-tagging* (i.e., labelling each word by its corresponding Part-Of-Speech tag such as noun or verb). We then use the POS-tagged tokens to define a chunk grammar to extract triplets of the form (subject, predicate, object).

Note that for the purpose of chunking, grammar is defined in such a way that it increases the chance of extracting proper subjects and objects from sentences, by ignoring auxiliary information about them. For instance for the sentence "*The small white cat went to the beautiful garden*", the words *cat* and *garden* will be recognized as the proper subject and object, respectively, by ignoring the further info on the size, color, and beauty of them in the sentence. Moreover, all components of the triplet will be represented by their original form, after *lemmatization*. For instance the triplet *(cat, go, garden)* is extracted from sentence above, where the verb went has been replaced by its lemmatized form *go*.

Also note that in the above approach, a chunk is defined by the sequence of $< NP >< VP >< NP >$, where *NP* denotes a noun phrase (such as the *small white cat* or *the beautiful garden*) and *VP* denotes a verb phrase (such as *went*).

Once all triplets of the above form are extracted from text corpus, the next step is to build a formal context that describes each subject and object in syntactical sense (here, *cat* and *garden*) by the verb associated to (here *go*). The method that we use for describing subjects/objects by their corresponding verb is the following. For a given triplet *(cat, go, garden)*, both subject and object in the syntactical sense (i.e., *cat* and *garden*) will be considered as two objects in the formal context (as two rows of the binary table) to be described by the verb go in the present *(going)* and past participle form *(gone)* as formal context attributes, respectively. In other words, *cat* is an object that has capability of *going* as an attribute, and *garden* is an object that is characterized by attribute *gone* which indicates it can be considered as a place that something (like *cat*) can *go* into. To formalize such definition, we define formal context attributes by adding suffix *-ing* and *-ed* to the lemmatized version of each verb in the existing triplets (here, *go-ing* and *go-ed*, although *go-ed* is meaningless from syntactical point of view). To ensure that the above explanation is well understood, we provide another example. For the sentence the big pharmaceutical enterprise *Roche* has been producing breast cancer medicine *Herceptin*, the triplet *(Roche, produce, Herceptin)* is translated into two additional rows in existing formal context as (1) *Roche* which has attribute produce *-ing* and (2) *Herceptin* which has attribute produce *-ed*. Once the formal context is generated (see Fig.2 as an example), we then use FCA to define formal concepts as described in the following section.

B. *From Meaningful Triplets to Concept Hierarchies Using Formal Concept Analysis (FCA)*

Formal Concept Analysis (FCA) [3], [4] helps in defining taxonomy in a given domain of interest and is being used for automatically defining formal concepts as well as learning concept hierarchies from a formal context. The formal context is described by a binary table (see Fig. 2), where each object (row) is characterized by a set of attributes (columns). Assume $A \subset X$ and $B \subset Y$ denote a subset of objects X and attributes Y, respectively. Then a pair (A , B) is a formal concept if and only if $A^u = B$ and $B^d = A$. Here, A^u denotes a set of those attributes that all objects in A have in common. Similarly, B^d denotes a set of those objects that all attributes in B have in common. A and B are known as extent and intent of that formal concept. We denote by $\{(A_1 , B_1) , (A_2 , B_2) , \dots , (A_p , B_p)\}$ the set of all p formal concepts embedded in a given formal context. To make story short, a formal concept is characterized by a subset of attributes that are common in a set of objects. Those formal concepts mimic the human interpretation of a real concept in the real world, yet represented in a mathematical form. Note that we may also refer to a formal concept (A,B) only by its extent A or intent B, or simply a name that is assigned to that pair. The concept lattice is then used for describing hierarchies between the formal concepts, i.e., sub-concept and super-concept relationships between formal concepts. For instance *mammals* (as a formal concept) can be considered as a sub-concept of a bigger formal concept labeled as *animals*. That means each creature (i.e., object) that belongs to the set of *mammals* also belongs to the set of *animals* but *not* vice versa.

	Hire-ing	Hire-ed	Produce-ing	Produce-ed	Study-ing	Study-ed	Sell-ing	Sell-ed	Collaborate-ing	Collaborate-ed
Roche	×		×				×		×	×
Herceptine				×		×		×		
HealthKit				×				×		
Apple	×		×				×			
John		×								
Genentech	×		×		×				×	×
Novartise	×		×		×		×			
Alex		×								
Clozapine				×		×		×		

Fig. 2 A formal context created by triplets of the form (subject, predicate, object). All subjects and objects of the sentences (in syntactical sense) are considered as objects (i.e., rows) of the formal context. Attributes of the formal context is then determined by the predicates (i.e., verbs) of the sentences or triplets. For each verb, there are two attributes: in the present form (i.e., *verb-ing*) and the past participle form (i.e., *verb-ed*). Subjects of the sentences are connected to verb-*ing*, and object of the sentences are linked to *verb-ed*.

There are many different algorithms proposed in the literature for extracting formal concepts [3], [7]{14], as well as concept lattices [15] , [16] , [17], [4], [18] , [19], [20] , [21], [7], [21] which are being used for transforming data into human understandable structures, knowledge representation [22] and data mining [23]. In this paper we use a tool called "*CONEXP*" for visualizing concept lattice.

In previous section we explained how triplets (subject, predicate, object) extracted from a text corpus can create a formal context. Once the formal context is created, then we apply FCA in order to build the concept hierarchy. We then use a reduced labelling method to simplify representation of formal concepts in the lattice. Reduced labelling ensures that each object x and attribute y of the formal context is represented only once in the lattice. Mathematically speaking, for each object $x \in X$ in the formal context, there is a formal concept (A,B) such that $\lambda(x) := (\{\{x\}^u\}^d , \{x\}^u) = (A , B)$. Similarly, for each attribute $y \in Y$, there is also a formal concept (C,D) such that $\mu(y) := (\{y\}^d, \{\{y\}^d\}^u) = (C,D)$. Here, $\lambda(.)$ and $\mu(.)$ denote *object concept* and *attribute concept* of x and y, respectively. The formal concepts (A,B) and (C,D) are then replaced by their corresponding object concepts and attribute concepts. Note that it is possible to have a subset of

objects x_1 , $x_{2,\dots,}x_n$ and attributes y_1 , $y_{2,\dots,}y_m$ that can be assigned to a given formal concept *(A,B)*. However, if x_1 is assigned to formal concept *(A,B)*, there is no formal concept *(C,D)* ≠ *(A,B)* such that x_1 can be assigned to, meaning that x_1 will be assigned to one and only one formal concept, i.e., one lattice node. Fig. 3 demonstrates the concept lattice that has been generated using FCA on the formal context that is depicted in Fig.2, but after reduced labelling.

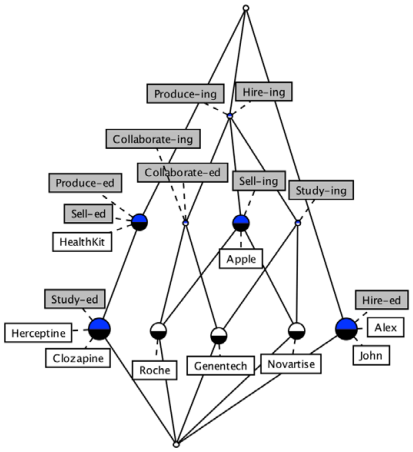

Fig. 3 The concept lattice that is generated after applying FCA to the formal context of Fig. 2. This figure depict the concept hierarchy after reduced labelling. By looking at this lattice, one can immediately recognize that objects such as *Roche*, *Novartis*, *Genentech*, and *Apple* belong to the same class (i.e., *companies*) which are in common in properties such as *hiring* and *producing*, but maybe different in some aspects such as *collaborating* and *studying*. Similarly, Objects such as *Herceptin*, *Clozapine*, and *HealthKit* are all instances of a same class (i.e., *products*) that are common in attributes such as *being produced*, *studied*, and *sold (i.e., sell-ed)*. *Alex* and *John* will also be recognized as a separate class (i.e., *persons*) that have a common characteristics as *being hired*. This example shows how such lattice can be of great help for learning about new terminologies. For instance, if someone knows *Herceptin* as a *product* (or *medication*) but has never heard about *Clozapine*, he can deduce that it should also be a *product* or *medication* and not a *company* or a *person*!

C. *From Concept Lattice to Ontology Using Relations and Domain Knowledge*

The first step towards learning ontology from a concept lattice is to define ontological concepts from formal concepts. The former can be interpreted as a label for the latter. For instance, verbs *{ hiring , producing , earning}*, as the intent of a formal concept, are characteristics of an ontological concept *company*, and objects *{ Roche , Novartis}* are just instances of that ontological concept. Similarly, the formal concept *{produced , sold , treating}* can be labeled as an ontological concept *medication*, where *{Herceptin , Perjeta}* are just instances of that concept. In order to build an ontology, we not only need ontological concepts, but also relationship between the concepts. This is not a difficult task because it can be done by connecting objects that are elements of the same triplets. For instance, objects *Roche* and *Herceptin* must be linked to each other because *(Roche , produce , Herceptin)* is an existing triplet already extracted from text corpus. The ontological relationships between *Roche* and *Herceptin* then can be defined as follows: *Roche – Produces –> Herceptin*, and *Herceptin – ProducedBy –> Roche*. Similarly all objects O_1 and O_2 for which there is a triplet *(O_1 , verb , O_2)* in the text corpus, will be linked to each other. For the implementation, we used *Apache Jena* for building ontology. Once the relationship between all objects is inserted into the lattice, the next step is to define *T-Box* using instances of *A-Box* already represented in the lattice. In other words, *(Roche , produce , Herceptin)* as well as other instances such as *(Novartis , produce , Clozapine)* are converted to the following *T-Box* instance: *(company , produce , medication)*. This is how a *T-Box* can be eventually constructed using instances of *A-Box*. Note that the domain knowledge and refinement of the concept lattice according to the comments from a subject matter expert can provide a great support for completing the ontology. In other words, the proposed method is just a starting point towards creation of an ontology from unstructured textual data and can further be adjusted according to a domain knowledge. Fig. 4 depicts the ontology presented in Protégé based on the concept lattice shown in Fig. 3.

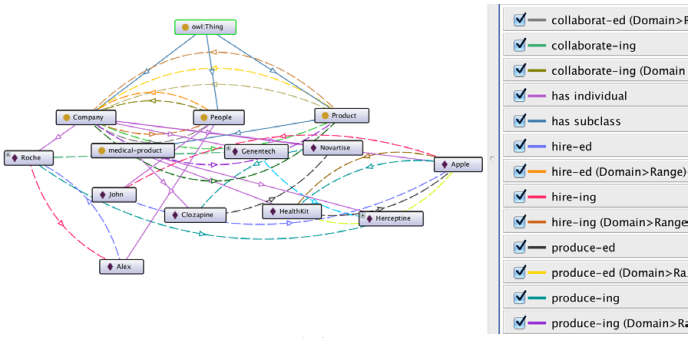

Fig. 4 The ontology created in Protégé based on the concept lattice shown in Fig. 3. The ontological concepts such as *company*, *people*, *product*, *medical-product*, etc. are linked to each other via relationships that have been defined based on the verbs in the triplets. The ontology is then be adjusted by adding further info such as the fact that *medical-product* is a sub-class of *product*. The description of the arrows and how they link different concepts to each other are also represented.

III. Results

We applied our proposed model to a given dataset which consists of 22 pdf files (2-6 pages each) containing information about news on companies and products in IT and pharmaceutical domain. After preparing the corpus and some preprocessing, we could identify 138 sentences from which we extracted triplets of the form (subject, predicate, object). This gave us a formal context with 123 objects (rows) and 40 attributes (columns). Note that the subjects, objects, and verbs in the triplets might be repeated more than once. This is why at the end, we ended up with a smaller formal context. We also excluded some of the sentences that had no meaningful interpretation. Fig. 5 depicts the concept lattice generated by our proposed method on the dataset we described above. Although the graph looks complex, one can find interesting insights by looking at such lattice. The instances of the same category (such as products, diseases, companies, people, etc.) have been automatically appeared in more or less the same locations in the graph. This is because the instances of the same category share the same characteristics (i.e., attributes) as they appear quite frequently with the same kind of verbs. Moreover, grouping such instances facilitates learning *T-Box* which is essential for learning final ontology. Completing this graph with additional edges that indicate the relationship between ontological concepts can be used for deriving implication rules, and semantic reasoning.

IV. Discussion

Nowadays, new data are being produced more than any time before, and so many new scientific and non-scientific disciplines have been proposed to the globe. It is often the

case that someone wants to discover a new domain but he lacks a basic knowledge about that context.

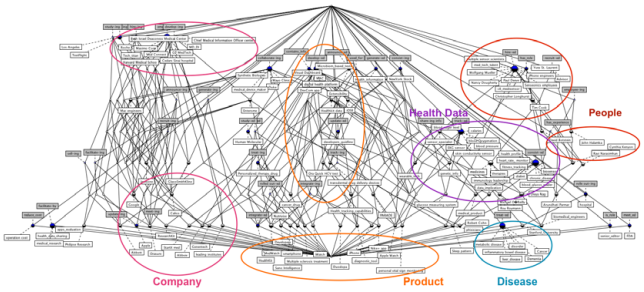

Fig. 5 Zoom in for readability. The concept lattice generated by our method using the documents on pharmaceutical and IT news. This corpus is made of 138 sentences from 22 pdf files (2-6 pages each). The corresponding formal context consists of 123 objects (rows) and 40 attributes (columns). The encircled nodes determine instances of similar concepts such as *companies (pink), products (orange), people(red)*, etc., which shares the same attributes (gray boxes). Note that a node in the lower layers of lattice inherits characteristics of its parents from upper layers (i.e., those through which it reaches to the root node indicating the most general concept).

Building ontologies from unstructured data is a good starting point for deriving a set of taxonomies and concept hierarchies for a given unknown domain. Besides building the cornerstone of a knowledge model, our proposed model can be used to classify new words into one of existing ontological concepts (or a perhaps a new one) depending on the characteristics or attributes that the new words share with existing ones (i.e., previously learned words). Moreover, we can use this method for indexing documents to be used by smart search engines (where documents are tagged by not only their own keywords, but also by other words that are somehow linked to the content of the documents in a semantic way). Another application of the proposed method for learning ontology is to derive implication rules and semantic knowledge. The possibility to further adjust the built ontology and to incorporate subject matter expert knowledge to come up with a more accurate ontology is also appealing. We would also like to mention some of the future works to be done for the purpose of improving our proposed method. The first topic is around NLP, and how to improve text preprocessing so that more meaningful triplets are extracted from the corpus. Dealing with huge number of documents is also another challenge. Using entity detection (e.g., which word is disease or medication) for building entity-specific ontologies (such as ontologies for diseases and medications, separately) and matching those ontologies via relational formal contexts (such as the one that links diseases to medications) also seems to be a right approach for learning more accurate and complete ontologies in a given domain of knowledge (such as medicine). As another example for future topic, we can also mention how to effectively update an ontology that has been created some time ago after receiving more documents in a given corpus. Adding more and more automation for handling big and complex graphs (as a consequence of a big corpus) is also a topic of interest for research.

V. Conclusion

We proposed a pipeline for generating ontology from a text corpus. This method is based on NLP and FCA, and gives structure to unstructured textual data. The application of the proposed method is to build a cornerstone for a knowledge model (especially when one has no clue about the context of available documents), which will be later used for semantic knowledge extraction and reasoning. The proposed method was described by a toy example (for simpler explanation) and was also tested on a real use case using a corpus of news documents (in pdf format). The underlying method was able to identify concepts and instances that are similar in sharing the common attributes. The new proposed approach for learning ontology from a text corpus gives a promising approach for generating knowledge models in a more automated way than before, which helps a lot in acquiring knowledge about a previously unknown context.

References

[1] Gobinda G Chowdhury. Natural language processing. Annual review of information science and technology, 37(1):51{89, 2003.

[2] James F Allen. Natural language processing. Encyclopedia of Cognitive Science, 2006J

[3] Bernhard Ganter and Rudolf Wille. Formal concept analysis: mathematical foundations. Springer Science & Business Media, 2012.

[4] Bernhard Ganter. Two basic algorithms in concept analysis. Springer, 2010.

[5] Philipp Cimiano, Andreas Hotho, and Ste_en Staab. Learning concept hierarchies from text corpora using formal concept analysis. J. Artif. Intell. Res.(JAIR), 24:305{339, 2005.

[6] Rokia Bendaoud, Amine Mohamed Rouane Hacene, Yannick Toussaint, Bertrand Delecroix, and Amedeo Napoli. Text-based ontology construction using relational concept analysis. In International Workshop on Ontology Dynamics-IWOD 2007, 2007.

[7] Lhouari Nourine and Olivier Raynaud. A fast algorithm for building lattices. Information processing letters, 71(5):199{204, 1999.

[8] Petr Krajca and Vilem Vychodil. Distributed algorithm for computing formal concepts using map-reduce framework. In Advances in Intelligent Data Analysis VIII, pages 333{344. Springer, 2009.

[9] Biao Xu, Ruairí de Fréin, Eric Robson, and Mícheál Ó Foghlú. Distributed formal concept analysis algorithms based on an iterative mapreduce framework. In Formal Concept Analysis, pages 292-308. Springer, 2012.

[10] Sergei O Kuznetsov. A fast algorithm for computing all intersections of objects in a finite semi-lattice. Automatic documentation and Mathematical linguistics, 27(5):11-21, 1993.

[11] Sergei O Kuznetsov. Learning of simple conceptual graphs from positive and negative examples. In PKDD, volume 99, pages 384-391. Springer, 1999.

[12] Petr Krajca, Jan Outrata, and Vilém Vychodil. Advances in algorithms based on cbo. In CLA, pages 325-337, 2010.

[13] Simon Andrews. In-close, a fast algorithm for computing formal concepts. 2009.

[14] Sergei O Kuznetsov and Sergei A Obiedkov. Comparing performance of algorithms for generating concept lattices. Journal of Experimental & Theoretical Artificial Intelligence, 14(2-3):189-216, 2002.

[15] Jean Paul Bordat. Calcul pratique du treillis de galois d'une correspondance. Mathematiques et Sciences humaines, 96:31-47, 1986.

[16] Claudio Carpineto and Giovanni Romano. Galois: An order-theoretic approach to conceptual clustering. In Proceedings of ICML, volume 293, pages 33-40, 1993.

[17] Michel Chein. Algorithme de recherche des sous-matrices premières d'une matrice. Bulletin mathématique de la Société des Sciences Mathématiques de la Réepublique Socialiste de Roumanie, pages 21-25, 1969.

[18] Robert Godin, Rokia Missaoui, and Hassan Alaoui. Learning algorithms using a galois lattice structure. In Tools for Artificial Intelligence, 1991. TAI'91., Third International Conference on pages 22-29. IEEE, 1991.

[19] Y Malgrange. Recherche des sous-matrices premières d'une matrice à coefficients binaires. applications à certains problèmes de graphe. In

Proceedings of the Deuxième Congrès de l'AFCALTI, pages 231-242, 1962.
[20] Eugene M Norris. An algorithm for computing the maximal rectangles in a binary relation. Revue Roumaine de Mathématiques Pures et Appliquées, 23(2):243-250, 1978.
[21] Bernhard Ganter and Sergei O Kuznetsov. Stepwise construction of the dedekind-macneille completion. In Conceptual framearc=Structures: Theory, Tools and Applications, pages 295-302. Springer, 1998.
[22] Rudolf Wille. Lattices in data analysis: How to draw them with a computer. Springer, 1989.
[23] Mohammed Javeed Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, Wei Li, et al. New algorithms for fast discovery of association rules. In KDD, volume 97, pages 283-286, 1997.

Simin JABBARI (Born in 26.07.1989 , Zanjan, IRAN) is a PhD student in Information Management Institute, University of Neuchatel, Neuchatel, Switzerland. She received her bachelor degree of Information Technology from the University of Zanjan (IRAN) and her master degree of Information System from University of Neuchatel (Switzerland). She is also doing her internship in F.Haffmann-La Roche Ltd, Kaiseraugst, Switzerland in “Dia Data Science Lab (DDSL)” group. Her research interests include knowledge representation, ontology , and data mining.

Prof. Dr. Kilian Stoffel is rector of University of Neuchatel and professor of computer science in the Information Management Institute (IMI) , Neuchatel (Switzerland). His main research interests include knowledge representation, machine learning ,ontologies, Information System and data mining.