Cheating to achieve Formal Concept Analysis over a large formal context

Victor Codocedo, Carla Taramasco, Hernan Astudillo

Cheating to achieve Formal Concept Analysis over a large formal context*

Victor Codocedo[1,3], Carla Taramasco[2], and Hernán Astudillo[1]

[1] Universidad Técnica Federico Santa María, Av. España 1640. Valparaíso, Chile.
[2] École Polytechnique, 32 Boulevard Victor 75015 Paris, France.
[3] LORIA, BP 70239, F-54506 Vandoeuvre-lès-Nancy, France.

Abstract. Researchers are facing one of the main problems of the *Information Era*. As more articles are made electronically available, it gets harder to follow trends in the different domains of research. Cheap, coherent and fast to construct knowledge models of research domains will be much required when information becomes unmanageable. While Formal Concept Analysis (FCA) has been widely used on several areas to construct knowledge artifacts for this purpose [17] (Ontology development, Information Retrieval, Software Refactoring, Knowledge Discovery), the large amount of documents and terminology used on research domains makes it not a very good option (because of the high computational cost and humanly-unprocessable output). In this article we propose a novel heuristic to create a taxonomy from a large term-document dataset using Latent Semantic Analysis and Formal Concept Analysis. We provide and discuss its implementation on a real dataset from the Software Architecture community obtained from the ISI Web of Knowledge (4400 documents).

1 Introduction

Research communities are facing one of the main problems of the Information Era and Formal Concept Analysis is not prepared to solve it. The amount of articles available online is growing each year yielding difficult to track trends, following ideas, looking for new terminology, etc. While some communities have understood the need for an artifact representing the knowledge within the domain (such as an ontology, a body-of-knowledge or a taxonomy) the problem remains in its construction since it is hard (highly technical), expensive (researchers are scarce) and complex (information is dynamic).

Automatic and semi-automatic creation of a terms taxonomy have been widely boarded in several fields [3,4,5,13,24]. In this work we focus on the approach described by Roth et al. [19] in which a taxonomy is derived from a corpus of documents by the use of Formal Concept Analysis (FCA). In particular, they describe an application used to *"represent a meaningful structure of*

* We would like to thank Chilean project FONDEF D08I1155 ContentCompass, intrabasal project FB/20SO/10 in the context of the Chilean basal project FB0821 and ECOS-CONICYT project C09E08 for funding this work.

a given knowledge community in a form of a lattice-based taxonomy". This application is illustrated using a set of abstracts of the embryologist community obtained from MedLine spanning 5 years where a random set of 25 authors and 18 terms were analyzed. Although the lattice-based taxonomy obtained was a fair representation of the domain, real-size corpora of research communities are rather much larger than this example.

Handling large datasets has been defined as one of the open problems in the community of FCA[4] for two main reasons: first, the computational costs involved in the calculation of the concept lattice can make the use of FCA prohibitive and second, the concept lattice structure yielded could be so complex that its use may be impossible [10].

Iceberg lattices [21] help in improving readability by eliminating *"not representative"* data, but useful information, such as *"emerging behaviors* [12,15], is lost in the process. Stabilized lattices (using a stability measure [16]) also improves readability by eliminating *"noisy elements"* from data, but being a post-process tool it also raises computational costs.

We describe in this document a novel heuristic to create a lattice-based taxonomy from a large corpus using Formal Concept Analysis and a widely used Information Retrieval technique called Latent Semantic Analysis (LSA). In particular, we describe a process to compress a *formal context* into a smaller *reduced context* in order to obtain a lattice of terms that can be used to describe the knowledge on a given research domain. We illustrate our approach using a real-size dataset from a research community of Computer Sciences.

The remainder of this paper proceeds as follows: Section 2 explains the basis of FCA, section 3 presents our approach and section 4, a case study over a real dataset from a research community. Section 5 presents the results and a comparison of the obtained taxonomy with a human-expert handmade thesaurus. Finally, the conclusions are described in section 6.

2 Formal Concept Analysis

Formal Concept Analysis, originally developed as a subfield of applied mathematics [23], is a method for data analysis, knowledge representation and information management. It organizes information in a lattice of formal concepts. A formal concept is constituted by its *extension* (the objects that compose the concept) and its *intension* (the attributes that objects share). Objects and attributes are placed as rows and columns (resp.) in a cross-table or *formal context* where each cell indicates whether the object of that row have the attribute of that column. In what follows, we describe the Formal Concept Analysis framework as synthesized by Wille [22].

[4] `http://www.upriss.org.uk/fca/problems06.pdf`

2.1 Framework

Let G be a set of objects, M a set of attributes and I a binary relation between G and M ($I \subseteq (G \times M)$) indicating by gIm that the object g contains the attribute m and $\mathbb{K} = (G, M, I)$ be the *formal context* defined by G, M and I. For $A \subseteq G$ and $B \subseteq M$ it is defined the *derivation operator* ($'$) as follows:

$$A' = \{m \in M \mid gIm, \forall g \in A\}, \text{ with } A \subseteq G \tag{1}$$

$$B' = \{g \in G \mid gIm, \forall m \in B\}, \text{ with } B \subseteq M \tag{2}$$

A *formal concept* of the *formal context* $\mathbb{K}$ is defined by (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$, where A is called the *extent* and B is called the *intent* of the concept. The set of all formal concepts is defined as $L(G, M, I)$.

For two formal concepts $(A_1, B_1), (A_2, B_2) \in \mathbb{K}$, the *hierarchy* of concepts is given by the relation subconcept-superconcept as follows:

$$(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2 (\iff B_1 \supseteq B_2) \tag{3}$$

Where $(A_1, B1)$ is called the *subconcept* and (A_2, B_2) is called the *superconcept*.

$\mathfrak{B}(\mathbb{K}) = (L(G, M, I), \leq)$ is the complete lattice or *concept lattice* of context $\mathbb{K}$

2.2 Iceberg Concept Lattices

Let (A, B) be a concept of $\mathfrak{B}(\mathbb{K})$, its *support* is defined as:

$$supp(A, B) = \frac{|A|}{|G|} \tag{4}$$

Given a threshold **minsupp** $\in [0, 1]$, the concept (A, B) is called a *"frequent concept"* if $supp(A, B) \geq$**minsupp**.

An **Iceberg lattice** [21] is the set of all frequent concepts for a given **minsupp**.

2.3 Stability

Stability was proposed by Kuznetsov in [14,16] as a mechanism to prune *"noisy concepts"*. It was extended by Roth et al. We use and provide their definition from [19] and [15]:

Let $\mathbb{K} = (G, M, I)$ be a formal context and (A, B) be a formal concept of $\mathbb{K}$. The stability index, σ, of (A, B) is defined as follows:

$$\sigma(A, B) = \frac{|\{C \subseteq A \mid C' = B\}|}{2^{|A|}} \tag{5}$$

Stability measures how much the intent of a concept depends on particular objects of its extent, meaning that if the formal context changes and some objects disappear, then stability indicates how likely it is for a concept to remain in the concept lattice. Stability can also be used to construct a *stabilized lattice* for a given threshold similarly to an *iceberg lattice*.

Analogous to definition 5, the **extensional stability** of a concept (A, B) can be defined as:

$$\sigma_e(A, B) = \frac{|\{D \subseteq B \mid D' = A\}|}{2^{|B|}} \tag{6}$$

Extensional stability measures how likely is for a concept to remain if some attributes are eliminated from the context. We will use both definitions in this work differentiating them as *intensional stability* (on (5)) and *extensional stability* (on (6)).

3 Reducing a large formal context

Different from Roth's approach [19], we are not interested in tracking groups of people working on groups of topics, but rather in the relations among topics. These relations occur in the articles that authors write, where topics or terms can appear in sets and each one can appear one or more times. To elaborate:

Given a corpus of articles G, a list of terms M and the relation among them $I \subseteq (G \times M)$ indicating by gIm that the article g contains the term m, the document-article formal context is defined as:

$$\mathbb{K}_O = (G, M, I) \tag{7}$$

3.1 Rationale

Even for a small set of terms, the amount of articles for a small research community can reach thousands of articles making the processing of $\mathbb{K}_O$ impossible or useless. The problem gets worse over time, because it can be expected that each year hundreds of articles will be added to the corpus.

What happens with terms over time? In taxonomy evolution, as described in [18], symmetric patterns arise: some fields will *progress or decline*; some fields will contain more or less concepts (*enrichment or impoverishment*); and some fields will *merge or split*. In any case, it is not expected that the amount of terms would vary greatly.

Latent Semantic Analysis (LSA) or Latent Semantic Indexing (LSI) [6] is a technique used commonly in Information Retrieval (IR) as a tool for indexation, clusterization and query answering. LSA is based on the idea that for a given set of terms and documents, *the relation among terms can be explained by a set of dimensions whose size is much smaller than the amount of documents*. We exploit this feature of LSA to construct a **reduced formal context of dimensions and terms** having as conditions that information regarding relations of

terms cannot be lost and that it has to produce a coherent taxonomy using less computational time. In what follows, we provide a brief description of LSA to elaborate on how we used it to produce a *reduced formal context*. For further reading, please refer to [6].

3.2 Latent Semantic Analysis

Given a list of m terms and a corpus of n documents, let A be a term-document matrix of rank-min(m,n) as defined in 8, where a_{ij} is the weight[5] of the term i in the document j. The Single-Value Decomposition of matrix A (in equation (9)) produces its factorization in three matrices where Σ contains the single-values of matrix A at the diagonal in descending order and the columns of matrices U and V are called left and right singular vectors of A.

$$A_{m\times n} = [a_{ij}] \; ; \; i = [1..m], j = [1..n] \tag{8}$$

$$A_{m\times n} = U_{m\times m} \cdot \Sigma_{m\times n} \cdot V^T_{n\times n} \tag{9}$$

$$A'_{m\times n} = U_{m\times k} \cdot \Sigma_{k\times k} \cdot V^T_{k\times n} \tag{10}$$

Since singular values drops quickly, we can create a new approximation of matrix A using $k \ll min(m,n)$ as shown in (10). Matrix $A' \approx A$ is the closest k-rank matrix approximation to A by the Frobenius measure [11]. Two new matrices can be calculated:

$$B_{m\times k} = U_{m\times k} \cdot \Sigma_{k\times k} \tag{11}$$

$$C_{n\times k} = V_{n\times k} \cdot \Sigma_{k\times k} \tag{12}$$

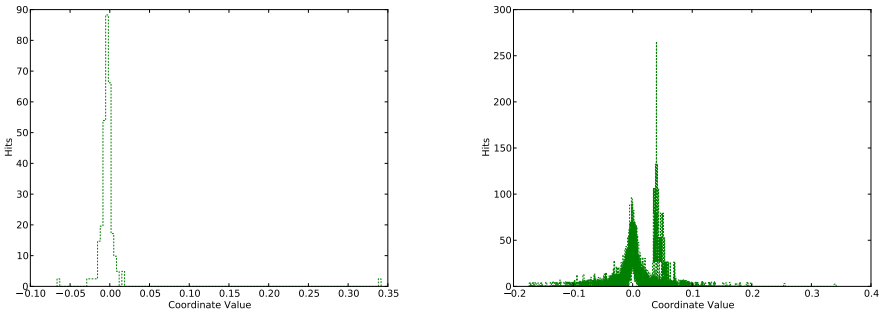
where B holds the vector-space representation in k dimensions of terms; and C the one of documents. Both of these matrices are used for clusterization since, on them, similar elements are closer on each dimension. In particular, each dimension on B (each column) has a Gaussian-like distribution where terms group around the mean value (see figure 1), except for dimension 0 (the different behavior in figure 1(b)) where terms have almost the same value[6]. We exploit this feature to define a *conversion-function* that allows us to construct the *reduced context*.

3.3 A probabilistic-based *conversion-function*

Which terms are related within a given dimension? Since each dimension holds continuous values, it is hard to define a region for them. Nevertheless, we know

[5] Several weighting functions can be used, being the most used frequency of term and term frequency-inverse document frequency (tf.idf)

[6] We do not use the information in this dimension for our analysis and exclude it from our results.



(a) Distribution of values in dimension 1 (b) Distribution of values in all dimensions

Fig. 1. Distribution of Dimensions' values in matrix B

that such a region has to be centered at the mean value of the dimension. Hence, we define a *"belonging region"* centered at the mean with a modifiable width. Terms in this region are related because they *belong* in the dimension and hence, the pair dimension-term will appear in the *reduced context*. The width of the *"belonging region"* is a parameter that allows us to manage the density of the context. The conversion function is defined as:

$$b_l(x,k) = \begin{cases} 1 & G_k(x) \in [\alpha, 1-\alpha] \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where function G_k is the probability density function (PDF) for dimension k and $\alpha \in]0{,}0.5[$ defines the limits of the *"belonging region"*.

3.4 Creating the reduced context

For a document-article formal context $\mathbb{K}_O$ as defined in (7) (*original context*) and a term-document matrix A as defined in (8) analogous to $\mathbb{K}_O$:

Given the factorization of matrix A as defined in (10), the vector-space representation of its terms in k dimensions B as defined in (11) and a conversion-function $b_l(x,k)$ as defined in (13):

Let D be the set of k dimensions in B

$$I_R \subseteq (D \times M) = \{(j,i) : \forall j \in D \wedge \forall i \in M \iff b_l(B_{ij}, j) = 1\} \tag{14}$$

we define the **reduced context** of $\mathbb{K}_O$ as $\mathbb{K}_R = (D, M, I_R)$.

Notice the inversion of pair (j,i) and B_{ij} performed to respect LSA conventions that require term-document matrices and FCA that uses document as objects and terms as attributes. In the *reduced context* we say *"dimension j contains term i if the evaluation of the conversion-function b_l over the value of the coordinate j for the term i is 1"*.

Summarizing, in order to get a *reduced context*, the values for α and k must be found.

3.5 Related approaches

Similar techniques have been proposed before. Gajdos et al[9] used LSA to reduce complexity in the structure of the lattice by eliminating noise in the formal context. While this approach is useful, it does not reduce the amount of data, but it *"tunes it"* to get a clearer result. Snasel et al. [20,9] proposed a matrix-reduction algorithms based on NMF. [7] and SVD[8]. While they state that these methods are successful to reduce the amount of concepts obtained using FCA, they do not describe a real life use of their technique (their experiment was performed over a 17x16 matrix) neither do they discuss about the performance of their approach. Kumar and Srinivas [1] approach consists of using fuzzy K-Means clustering [9] to reduce the attributes in a formal term-document context. In their approach, documents are categorized in k clusters using the *cosine similarity measure*. Cheung et al. [2] introduced *term-document* lattices complexity reduction by defining a set of equivalence relations that allows to reduce the set of objects. Finally, Dias et al. introduced JBOS [7] (junction based on objects similarity) which proposed a similar method where objects where group into prototype objects by calculating its similarity according to certain weights assigned manually to attributes.

4 Case Study: Software Architecture Community

The Software Architecture Corpus (SAC) was composed by extracting metadata from papers retrieved by the ISI Web of Knowledge search engine [10] using the query "software architecture". It is assumed that the keyword "software architecture" is present in each paper on their titles and/or abstracts.

While the search engine retrieved 4701 articles, not all of them have an abstract to work with. Those are excluded from our analysis leaving 4565 articles spanning from 1990 to 2009 (retrieved documents span from 1973 to 2009).

4.1 Term list

A term list was assembled by using Natural Language Processing over the articles' titles and abstracts. In order to avoid common words, a *stopword* list and a lexical tagger were used as a filter. A list of candidate terms was then manually filtered to obtain a final list of 120 terms, which included words and multi-words (such as *"Unified Model Language"*). Table 1 shows a sample of selected terms.

Each term was looked up on each document and its frequency of use was calculated. Then, a weighting measure was applied (tf.idf[11]) to each value. The

[7] Non-negative matrix factorization
[8] Single-Value Decomposition
[9] K-Means Clustering is a classic clustering technique for vector-space models
[10] http://isiwebofknowledge.com
[11] Term Frequency-Inverse Document Frequency is a weighting measure commonly used on IR based on the notion that term infrequency on a global scale makes it important.

Table 1. Top 10 more frequent terms

Term	Frequency
design	1710
development	1450
component	1253
process	1083
implementation	1006
datum	874
requirement	869
analysis	851
framework	817
control	801

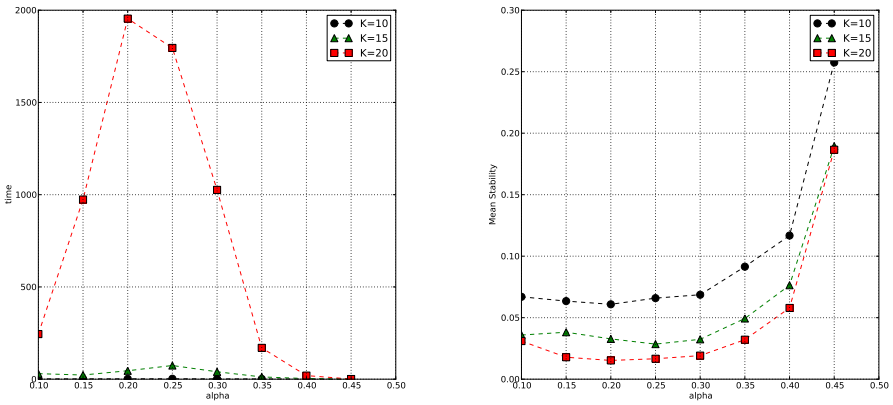
term-document matrix $A_w = a_{ij}$ was constructed using the final list of terms (M) and the corpus of documents (G) where a_{ij} represents the weight of term i in document j. We defined the relation $I \subseteq (G \times M) = \{(j, i) : \forall j \in G \wedge \forall i \in M \iff a_{ij} > 0\}$ to build up the **original context** $\mathbb{K}_O = (G, M, I)$ describing that a document *contains* a term only if its weight on it is over 0. The formal context $\mathbb{K}_O$ was used later to compare our reductions.

4.2 Reducing the SAC

As we stated at the end of section 3.3, two parameters had to be set in order to create the *reduced context*. Sadly, in LSA there is not a known method to find the best value for k, and not knowing that, it is not possible to find a good value for α. We defined a set of goals to observe which were the values of k and α that best accomplished them. The goals defined were:

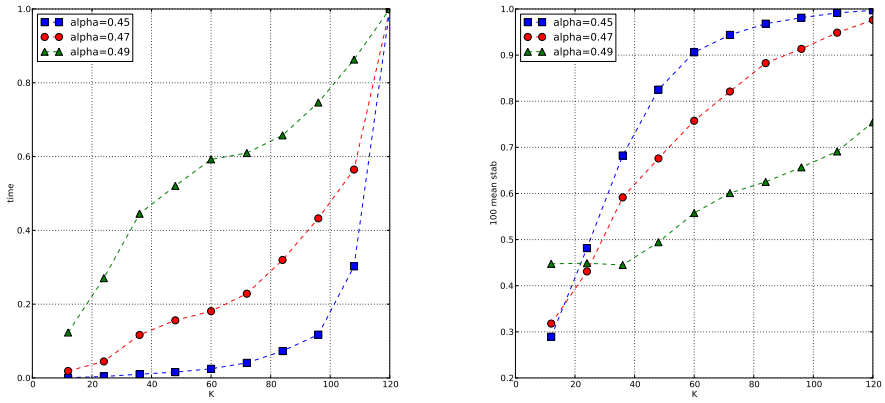
- Low Execution time
- High Stability
- Few Concepts in the final lattice

Using three fixed values for k we reduced several contexts and processed them through FCA in order to find the best value for α. As shown in figure 2, it was found that higher values of α (close to 0.5) yields the best results. Repeating the experience with 3 fixed values for α (0.45, 0.47 and 0.49) to find the best value for k we found a trade-off between stability and execution time as it can be observed in figure 3. Higher values of k yield higher stability but also a high execution time, and vice-versa. Since stability drops fast on k=60 and in the same value the execution time grows greatly, we selected it to obtain our results. α was set on 0.45 and 0.47.



(a) alpha vs Execution Time (b) alpha vs Stability

Fig. 2. Fixed K, Variable α



(a) K vs Execution Time Normalized (b) K vs Top 100 Mean Stability

Fig. 3. Variable K, Fixed α

5 Results and Discussion

Table 2 shows a comparative of the characteristics of the lattices yielded from two *reduced contexts* ($\mathbb{K}_R$) and the *original context* ($\mathbb{K}_O$). The lattices were processed using the FCA suite Coron System[12].

[12] `http://coron.loria.fr/`

Table 2. Comparison of characteristics

	$\alpha = 0,45$	$\alpha = 0,47$	Original
Objects	60	60	4565
Attributes	120	120	120
Density [%]	17,24	10,59	6,59
Concepts	6309	1207	170606
Coincidental Intents	3029	815	-
Mean attributes per concept	20,52	12,6	7,91
Intensional Stability	0,2170	0,3041	0.3995
Extensional Stability	0.2277	0.3211	0.1103
100 Top Int. Stab.	0,9061	0,7576	1
100 Top Ext. Stab.	0.9515	0.8287	0.9837
Levels	10	7	10
Time [s]	6,869	1,145	2865,723
Time to reduce [s]	39,333	39,325	-

Results shows that using LSA before FCA performs a **clear reduction in the formal context** from a size of 4565×120 (original context) to 60×120 (reduced context), specifically a reduction of 76 times the amount of data to be processed. It also **lowers the amount of concepts** yielded in the final lattice (27 and 141 times for α equal to 0.45 and 0.47 resp.), and because of that the **time required to calculate the full concept lattice is considerably reduced**, even considering the time required to create the *reduced contexts*.

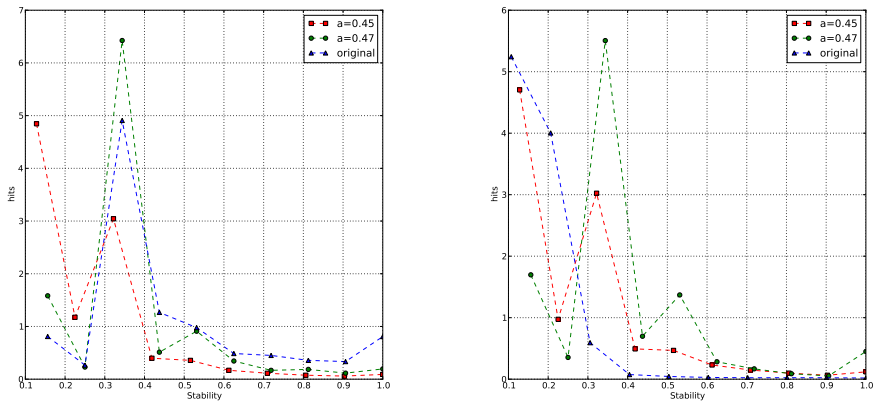
Stability gives more clues about the good quality of the reduction. Figure 4 shows intensional and extensional stability distribution. As it can be observed, the original context's lattice has a better intensional stability than the reduced contexts but a worst extensional stability. Mean values for these two measures are shown in table 2.

Since we have eliminated redundant data, each dimension is almost equally important meaning that in *reduced contexts* we cannot afford to eliminate a subset of them without affecting greatly the structure of the lattice obtained. In this case, we have eliminated a big part of the noise (k=60 was in fact a very good choice). On the other hand, the growth in extensional stability reflects that the structure of the reduced lattices is not tied to some specific terms. Some terms can be removed and the structure of the lattice would not vary greatly, which is what happens each year (see section 3.1).

5.1 A Software Architecture Taxonomy

Figure 5 shows the reduced notation of the lattice for the *reduced context* (k=60 and $\alpha = 0.45$). This lattice was drawn with Coron-drawer[13] a set of scripts specially written for large lattices. For the sake of space and simplicity we provide

[13] `http://code.google.com/p/coron-drawer/`



(a) Intensional Stability (b) Extensional Stability

Fig. 4. Stability distribution (k=60)

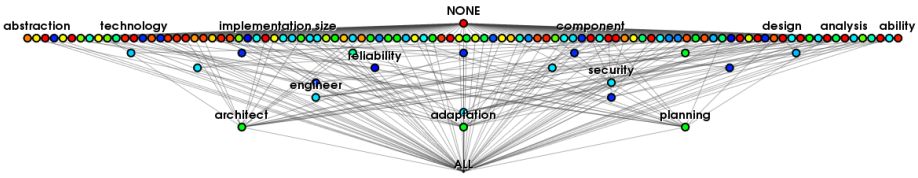


Fig. 5. Filtered Lattice (K=60, $\alpha = 0.45$, minsupp=0)

a small comparison of the terms in the reduced lattice-based taxonomy with a human-expert handmade thesaurus of Software Architecture [8].

Software Architecture Thesaurus Comparison The thesaurus contains 494 elements (we call them elements to differentiate them from lattice's concepts and taxonomy's terms) organized in a hierarchical fashion. They have at most one parent and the hierarchy has multiple roots. The thesaurus is exhaustive and comprises mainly definitions of Software Architecture's concepts and entities (such as framework's names or important authors in the domain). The comparison shows:

– From our 120 term list, 50 terms (42%) match exactly with a term on the thesaurus. 25 terms (21%) have a semi-match, meaning that they are part of a term on the thesaurus (*database* in our hierarchy and *shared database* in the thesaurus) and 45 (37%) terms do not have a simile in the thesaurus.

– The three main concepts *design, analysis and framework* (with support over 50%) found in our taxonomy, also remain being main elements in the thesaurus.
– Even when some elements in the thesaurus are not found in our taxonomy, they actually exists as relations among terms. For instance, the relation among the terms *design* and *pattern* describe the thesaurus' element *design pattern*. This is also true for *design decision*, *information view*, *knowledge reuse*, *quality requirements*, *business methodology* and several more elements.

6 Conclusions

In this work we have presented a method and a technique to apply Formal Concept Analysis (FCA) to large contexts of data in order to obtain a lattice-based taxonomy. We have outlined that large-size datasets are not suitable to be processed by FCA and that, this fact is an important problem in the domain.

The solution presented here, is based on an Information Retrieval technique called Latent Semantic Analysis which is used to reduce a term-document matrix to a much smaller matrix where terms are related to a set of dimensions instead of documents. Using a probabilistic approach, this matrix is converted into a binary formal context where FCA can be applied.

The approach was illustrated with a case study using a research domain from computational sciences called *Software Architecture*. The corpus created for this domain consists of more than 4500 documents and 120 terms. We have compared the characteristics of the lattice obtained through FCA from the original formal context of terms and documents and the reduced contexts generated by our approach. We have found that not only our approach is considerably more economic in execution time as well as in the amount of concepts obtained in the final lattice but intensional and extensional stabilities give us elements to be certain of the quality of our approach.

A small comparison with a human expert handmade thesaurus of the community of Software Architecture is provided in order to illustrate that a real and coherent taxonomy can be obtained using our approach.

References

1. Ch. Aswani Kumar and S. Srinivas. Concept lattice reduction using fuzzy K-Means clustering. *Expert Systems with Applications*, 37(3):2696–2704, March 2010.
2. Karen S. K. Cheung and Douglas Vogel. Complexity Reduction in Lattice-Based Information Retrieval. *Information Retrieval*, 8(2):285–299, April 2005.
3. Philipp Cimiano, Andreas Hotho, and Steffen Staab. Learning concept hierarchies from text corpora using formal concept analysis. *J. Artif. Int. Res.*, 24:305–339, August 2005.
4. Víctor Codocedo and Hernán Astudillo. No mining, no meaning: relating documents across repositories with ontology-driven information extraction. In *Proceeding of the eighth ACM symposium on Document engineering*, DocEng '08, pages 110–118, New York, NY, USA, 2008. ACM.

5. Wisam Dakka, Panagiotis G. Ipeirotis, and Kenneth R. Wood. Automatic construction of multifaceted browsing interfaces. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 768–775, New York, NY, USA, 2005. ACM.
6. Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the american society for Information Science*, 41(6):391–407, 1990.
7. Sergio M. Dias and Newton J. Vieira. Reducing the size of concept lattices: The JBOS Approach. In *Proceedings of the 8ht international conference on Concept Lattices and their Applications*, CLA 2010, pages 80–91, 2010.
8. Anabel Fraga and Juan Lloréns. Training initiative for new software/enterprise architects: An ontological approach. In *WICSA*, page 19. IEEE Computer Society, 2007.
9. Petr Gajdos, Pavel Moravec, and Václav Snásel. Concept lattice generation by singular value decomposition. In Václav Snásel and Radim Belohlávek, editors, *International Workshop on Concept Lattices and their Applications (CLA)*, volume 110 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.
10. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin/Heidelberg, 1999.
11. Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.
12. Nicolas Jay, François Kohler, and Amedeo Napoli. Analysis of social communities with iceberg and stability-based concept lattices. In *Proceedings of the 6th international conference on Formal concept analysis*, ICFCA'08, pages 258–272, Berlin, Heidelberg, 2008. Springer-Verlag.
13. John Kominek and Rick Kazman. Accessing multimedia through concept clustering. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '97, pages 19–26, New York, NY, USA, 1997. ACM.
14. Sergei Kuznetsov. Stability as an estimate of the degree of substantiation of hypotheses derived on the basis of operational similarity. nauchn. tekh. inf., ser.2 (automat. document. math. linguist.). 12:21–29, 1990.
15. Sergei Kuznetsov, Sergei Obiedkov, and Camille Roth. Reducing the representation complexity of lattice-based taxonomies. In Uta Priss, Simon Polovina, and Richard Hill, editors, *Conceptual Structures: Knowledge Architectures for Smart Applications*, volume 4604 of *Lecture Notes in Computer Science*, pages 241–254. Springer Berlin / Heidelberg, 2007.
16. Sergei O. Kuznetsov. On stability of a formal concept. *Annals of Mathematics and Artificial Intelligence*, 49:101–115, April 2007.
17. Uta Priss. Formal concept analysis in information science. *Annual Review of Information Science and Technology*, 40(1):521–543, September 2007.
18. Camille Roth and Paul Bourgine. Lattice-based dynamic and overlapping taxonomies: The case of epistemic communities. *SCIENTOMETRICS*, 69(2):429–447, NOV 2006.
19. Camille Roth, Sergei Obiedkov, and Derrick Kourie. Towards concise representation for taxonomies of epistemic communities. In *Proceedings of the 4th international conference on Concept lattices and their applications*, CLA'06, pages 240–255, Berlin, Heidelberg, 2008. Springer-Verlag.
20. Vaclav Snasel, Martin Polovincak, and Hussam M. Dahwa. Concept lattice Reduction by Singular Value Decomposition. *Proceedings of the Spring Young Researcher's Colloquium on Database and Information Systems*, 2007.

21. Gerd Stumme. Efficient data mining based on formal concept analysis. In Abdelkader Hameurlain, Rosine Cicchetti, and Roland Traunmüller, editors, *Database and Expert Systems Applications*, volume 2453 of *Lecture Notes in Computer Science*, pages 3–22. Springer Berlin / Heidelberg, 2002.
22. Rudolf Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In Ivan Rival, editor, *Ordered sets*, pages 445–470, Dordrecht–Boston, 1982. Reidel.
23. Rudolf Wille. Formal concept analysis as mathematical theory of concepts and concept hierarchies. In Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors, *Formal Concept Analysis*, volume 3626 of *Lecture Notes in Computer Science*, pages 1–33. Springer Berlin / Heidelberg, 2005.
24. Jian-hua Yeh and Naomi Yang. Ontology construction based on latent topic extraction in a digital library. In George Buchanan, Masood Masoodian, and Sally Cunningham, editors, *Digital Libraries: Universal and Ubiquitous Access to Information*, volume 5362 of *Lecture Notes in Computer Science*, pages 93–103. Springer Berlin / Heidelberg, 2008.