Conceptual Clustering of Text Clusters

Andreas Hotho, Gerd Stumme[*]

[*] Institute of Applied Informatics and Formal Description Methods AIFB, University of Karlsruhe, D–76128 Karlsruhe, Germany; http://www.aifb.uni-karlsruhe.de/WBS, {hotho, stumme}@aifb.uni-karlsruhe.de

Abstract. Common clustering techniques have the disadvantage that they do not provide intensional descriptions of the clusters obtained. Conceptual Clustering techniques, on the other hand, provide such descriptions, but are known to be rather slow. In this paper, we discuss a way of combining both techniques. We first cluster the documents by a variant of k–Means, using a thesaurus as background knowledge. This clustering reduces the large number of documents to a relatively small number of clusters, which can then be clustered *conceptually* in the second step.

Keywords. Text Clustering, Conceptual Clustering, k–Means, Formal Concept Analysis

1 Introduction

Common clustering techniques have the disadvantage that they do not provide intensional descriptions of the clusters obtained. Conceptual Clustering techniques, on the other hand, provide such descriptions, but are known to be rather slow. In this paper, we discuss a way of combining both techniques.

Our approach consists of two steps. First, we apply a common (non-conceptual) clustering algorithm — in our case a variant of the well-known k–Means algorithm — in order to decrease the size of the problem. Then we cluster the resulting clusters using a conceptual clustering technique — in our case, Formal Concept Analysis. The latter provides intensional descriptions of the resulting clusters; and is efficient enough, if the number of clusters chosen in the first clustering step is not too high. The resulting concept lattice can then be accessed using existing techniques from Formal Concept Analysis.

In this paper, we focus on the problem of text clustering. In order to improve the quality and understandibility of the clusters, we additionally make use of background knowledge in form of a thesaurus. In our application, we used WordNet.

The problem addressed can thus be described as follows: Given a set of documents and a thesaurus, provide a clustering of the documents with reasonable performance, which comes along with intensional descriptions of the clusters.

In this paper, we discuss our approach along the Reuters–21578 text collection. The remainder of the paper is organized as follows. In the next section, we describe the Reuters data, the preprocessing we performed, the non-conceptual clustering step, and the extraction of cluster descriptions. In Section 3, we recall the basic notions of Formal Concept Analysis, explain how the document clusters are clustered conceptually, and discuss the results. Section 4 provides an overview over related work. At the end of the paper, we discuss some future research issues.

2 Clustering the documents

In this section we describe the dataset we used for the evaluation, and the non-conceptual clustering part. This part consists of the preprocessing of documents, mapping words to synsets of WordNet, the non-conceptual clustering itself, and the extraction of cluster descriptions. The purpose of this first clustering step is to reduce the number of objects (and the number of describing attributes) so that it can be treated in reasonable time by Formal Concept Analysis. The final output of this part will be a set of clusters, together with a list of terms for each cluster describing it best.

In this paper, we will use the expression ‘term’ both for words and for terms (synsets) of the thesaurus for sake of simplicity. If we talk about one of them specifically, we will mention it explicitly.

2.1 The Reuters-21578 Dataset

We selected the Reuters-21578[1] text collection for our experiments. The corpus consists of 21578 documents. This corpus is especially interesting for evaluation, as part of it comes along with a (hand-crafted) classification. It contains 135 so-called topics. To be more general, we will refer to them as ‘classes’ in the sequel. For allowing evaluation, we restrict ourselves to the 12344 documents which have been classified manually by Reuters. Some of them could not be assigned by the experts to one of the predefined classes; we collect them in an additional class ‘defnoclass’. In order to make the problem more homogeneous, we drop all classes with less than 25 documents and randomly prune documents in each class to at most 30 documents. Reuters assigns some of its documents to multiple classes, but we consider only the first assignment. After these steps, we obtain our final corpus $\mathcal{D}$ for evaluation. It consists of 1015 documents, distributed over 34 Reuters topics.[2]

2.2 Preprocessing the Document Set

For the preprocessing of the documents, we used the text mining system developed at AIFB within the KAON[3] framework. We performed the following steps on the selected corpus: First we lowered the letters of all words and removed stopwords. We used a stopword list with 571 entries which removed 374 stopwords from the documents. We also dropped all words with less than five occurrences over the whole corpus. 4257 words were removed in total. After these steps, 2311 different words remained in our list, with a total occurrence of 85284.

2.3 WordNet as Background Knowledge

Instead of using a bag-of-word model directly, we additionally enriched it with background knowledge. The idea was to replace the words by terms and their broader terms of a given thesaurus, in order to capture also similarities on a higher conceptual level. For this purpose we needed a resource suitable for the Reuters corpus. We choose WordNet[4] as our background knowledge. WordNet consists of so-called synsets, together with a hypernym/hyponym hierarchy.[5]

First we replaced all nouns appearing in the documents with synsets from WordNet (and omitted the rest). As the assignment of words to synsets is ambiguous, we implemented several strategies. The strategy we finally used was to assign to each word the synset that WordNet suggests as the most probable.

Then we used the hypernym/hyponym hierarchy on the synsets of WordNet to add more general terms, which later help identifying related topics that are addressed by (seemingly) different words. We added to each synset its four most specific hypernyms. The number of four was chosen for not obtaining too general (and hence non-distinguishing) terms. The synsets that were assigned to at least one document formed then the set $\mathcal{T}$ of terms, which is used for describing the documents.

We performed our approach also without thesaurus. In order to capture the declination of nouns (which was implicitly done by the mapping to synsets in the approach described above), we applied a Porter Stemmer [14]. It showed that the thesaurus-based approach performed better in terms of accuracy, hence we dropped the stemming approach.

2.4 Building the Term Vectors

Based on the work done so far, we built a term vector for each document $d \in \mathcal{D}$. For each document, the terms $t \in \mathcal{T}$ are weighted by *tfidf* (term frequency × inverse document frequency) [15], which is defined as follows:

$$\mathit{tfidf}(d,t) = \mathit{tf}(d,t) \times \log\left(\frac{|\mathcal{D}|}{|\mathcal{D}_t|}\right) \qquad (1)$$

where $\mathit{tf}(d,t)$ is the frequency of term t in document d, and $\mathcal{D}_t \subseteq \mathcal{D}$ is the set of all documents containing term t. The term vector for document d is then the tuple $\vec{w}_d := (\mathit{tfidf}(d,t))_{t \in \mathcal{T}}$.

Tfidf weighs the frequency of a term in a document with a factor that discounts its importance when it appears in almost all documents. Therefore terms that appear too rarely or too frequently are ranked lower than terms that hold the balance and, hence, are expected to be better able to contribute to clustering results.

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

[2] acq, alum, bop, carcass, cocoa, coffee, copper, cotton, cpi, crude, defnoclass, dlr, earn, gas, gnp, gold, grain, interest, ipi, iron-steel, jobs, livestock, money-fx, money-supply, nat-gas, oilseed, pet-chem, reserves, rubber, ship, sugar, tin, trade, veg-oil

[3] http://kaon.semanticweb.org

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

[5] See http://www.cogsci.princeton.edu/~wn/man1.7.1/ wn-gloss.7WN.html

2.5 Clustering the Documents with BiSec–k–Means

On the preprocessed data (as described in 2.2) we applied a variant of k–Means, the 'bisecting' k–Means (in the following called BiSec–k–Means), using the so–called cosine similarity: We calculate the similarity between two documents $d_1, d_2 \in \mathcal{D}$ as the cosine of their word vectors $\vec{w}_1$ and $\vec{w}_2$, which can be computed as follows:

$$\cos(\sphericalangle(\vec{w}_1, \vec{w}_2)) = \frac{\vec{w}_1 \cdot \vec{w}_2}{\| \vec{w}_1 \| \cdot \| \vec{w}_2 \|} = \frac{\sum\limits_{t \in \mathcal{T}} (\mathit{tfidf}(d_1, t) \cdot \mathit{tfidf}(d_2, t))}{\sqrt{\sum\limits_{t \in \mathcal{T}} \mathit{tfidf}(d_1, t)^2} \cdot \sqrt{\sum\limits_{t \in \mathcal{T}} \mathit{tfidf}(d_2, t)^2}} \tag{2}$$

For the non-conceptual clustering step we need a fast algorithm (such as k–Means) to deal with large datasets, which should also provide a reasonable accuracy. Instead of a slow agglomerative clustering technique with a good accuracy we choose BiSec–k–Means which tends to give better results as k–Means and is sometimes also better as agglomerative clustering, while it is as fast as k–Means (cf. [16]). BiSec–k–Means is based on the k–Means algorithm, which works as follows:

> Let k be the number of desired clusters.
>
> - Choose randomly k points as starting centroids.
> - Assign each point to the closest centroid (with respect to a given similarity measure).
> - (Re-)calculate all cluster centroids.
> - Repeat the last two steps until the centroids do not change any more.

BiSec–k–Means applies k–Means repeatedly $k_b - 1$ times where k_b is the predefined number of clusters for BiSec–k–Means. At the first time, k–Means is performed for $k = 2$. Then the cluster with the highest cardinality is selected and split into two new clusters; using again k–Means with $k = 2$. This procedure is repeated until the requested k_b clusters are built. The set of clusters is denoted by $\mathcal{C}$. The centroid of a cluster $C \in \mathcal{C}$ is denoted by $\vec{w}_C$.

This clustering reduces the large number of documents to a relatively small number of clusters, which can then be clustered *conceptually* in the second step. The idea is, however, to keep that number as large as possible, since the more of the clustering is done conceptually, the better the results will be interpretable.

For clustering the obtained document clusters conceptually, we need a description for each of the clusters. We describe next, how these descriptions are extracted.

2.6 Extracting Cluster Descriptions

For applying a conceptual clustering approach like Formal Concept Analysis (FCA), we need intensional descriptions of the objects to be clustered. In our scenario this means that we have to decide, for each thesaurus term and each cluster, if the term shall be considered as being important for the cluster or not. For performance reasons, we also would like to keep the total number of selected terms small.

Therefore we need a method which points us to the most important terms for each cluster. We introduce a threshold θ to decide whether a term is important or not. This way we are also able to control how many terms remain to describe the clusters. In our application, we used a threshold of 25 % of the maximal value.

We used the centroid vectors of the clusters for extracting the cluster descriptions. For each cluster, the description of the cluster is the set of all terms having a value in the centroid vector which is above the threshold θ. This assures that those terms are selected which are most important for the cluster. All terms which were not assigned to at least one cluster were finally dropped. The resulting set is denoted by $\mathcal{T}_c$. The assignment of the terms to the clusters is the basis for the next step, the conceptual clustering part.

3 Conceptual Clustering of the Document Clusters

Now we consider the clusters of documents as atomic objects which will be clustered conceptually. As the number of objects is thus reduced to a 'reasonable' size, we are able to apply a conceptual clustering technique. We will obtain a clustering of document clusters, where each cluster of document clusters comes along with an intensional description. This description then serves also as description of the documents themselves.

3.1 Conceptual Clustering by Formal Concept Analysis

As conceptual clustering technique, we make use of Formal Concept Analysis. Formal Concept Analysis (FCA) was introduced as a mathematical theory modeling the concept of 'concepts' in terms of lattice theory. We recall the basics of Formal Concept Analysis (FCA) as far as they are needed for this paper. An extensive overview is given in [5]. To allow a mathematical description of concepts as being composed of extensions and intensions, Formal Concept Analysis starts with a *formal context*:

Definition: A *formal context* is a triple $\mathbb{K} := (G, M, I)$, where G is a set of *objects*, M is a set of *attributes*, and I is a binary relation between G and M (i. e. $I \subseteq G \times M$). $(g, m) \in I$ is read "*object g has attribute m*".

In our application, the set of objects consists of all clusters determined in the previous step, i. e., $G := \mathcal{C}$. The set of attributes consists of all terms which remain from the step described in Section 2.6, i. e., $M := \mathcal{T}_c$; and the relation I indicates if a term is related to a cluster, i. e., if its value in the centroid vector is above the threshold θ: $(C, t) \in I :\iff (\vec{w}_C)_t \geq \theta$. In the sequel, 'attribute' and 'term' are thus used synonymously. 'Object' is used synonymously with 'BiSec–k–Means cluster' unless otherwise stated.

From a formal context, a concept hierarchy, called *concept lattice*, can be derived:

Definition: For $A \subseteq G$, we define $A' := \{m \in M \mid \forall g \in A\colon (g, m) \in I\}$ and, for $B \subseteq M$, we define $B' := \{g \in G \mid \forall m \in B\colon (g, m) \in I\}$.

A *formal concept* of a formal context (G, M, I) is defined as a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. The sets A and B are called the *extent* and the *intent* of the formal concept (A, B). The *subconcept–superconcept relation* is formalized by

$$(A_1, B_1) \leq (A_2, B_2) :\iff A_1 \subseteq A_2 \quad (\iff B_1 \supseteq B_2) \ .$$

The set of all formal concepts of a context $\mathbb{K}$ together with the partial order $\leq$ is always a complete lattice,[6] called the *concept lattice* of $\mathbb{K}$ and denoted by $\underline{\mathfrak{B}}(\mathbb{K})$.

3.2 Visualizing the Concept Hierarchy

Figure 1 highlights a part of the concept lattice of our context by a *line diagram*. It will be explained in detail below. The lattice was computed and visualized using the Cernato software of NaviCon Gmbh.[7]

Line diagrams follow the conventions for the visualization of hierarchical concept systems as established in the international standard ISO 704. In a line diagram, each node represents a formal concept. Due to technical reasons, we reverse the usual reading order: A concept $\mathfrak{c}_1 \in \underline{\mathfrak{B}}(\mathbb{K})$ is a subconcept of a concept $\mathfrak{c}_2 \in \underline{\mathfrak{B}}(\mathbb{K})$ if and only if there is a path of ascending(!) edges from the node representing $\mathfrak{c}_2$ to the node representing $\mathfrak{c}_1$. The name of an object g is always attached to the node representing the most specific concept (i. e., the smallest concept with respect to $\leq$) with g in its extent (i. e., in our figure, the highest such node); dually, the name of an attribute m is always attached to the node representing the most general concept with m in its intent (i. e., the lowest such node in the diagram). We can always read the context relation from the diagram, since an object g has an attribute m if and only if the concept labeled by g is a subconcept of the one labeled by m. The extent of a concept consists of all objects whose labels are attached to subconcepts, and, dually, the intent consists of all attributes attached to superconcepts.

For example, the concept labeled by 'refiner' has $\{$CL 1, CL 3$\}$ as extent, and $\{$(h)refiner, (h)oil, $\ldots$, '(h)compound, chemical compound'$\}$ as intent. (h) indicates here WordNet synsets.

In the diagram, we can for instance see that there is a chain of concepts with increasing specificity. The most general of them (beside the top concept) contains in its extent clusters of documents addressing chemical compounds: CL 1, CL 3, CL 11, CL 17, and CL 33. In the next concept, they are restricted to document clusters related to oil: CL 1, CL 3, CL 11. The following concept considers only two of these clusters, namely the clusters 1 and 3. These are the only clusters talking about refining oil.

When we finally have a look at the attribute labels of the two concepts labeled by 'CL 1' and 'CL 3', resp., then we see that they address in fact different aspects of refining oil: The documents in Cluster 1 deal with the refinement of plant oil, while the documents in Cluster 3 have crude oil as subject.

The resulting concept lattice can also be interpreted as a concept hierarchy directly on the documents, as it is isomorphic to the concept lattice of the context $\mathbb{K}' := (G', M', I')$ with $G' := \mathcal{D}$, $M' := \mathcal{T}_c$, and $(d, t) \in I'$ iff $d \in C$ and $(\vec{w}_C)_t \geq \theta$ for some cluster $C \in \mathcal{C}$. This context is in fact an approximation of the descriptions of the documents by term vectors, with the property that all documents in one cluster obtain exactly the same description. This loss of information is the price we pay for improving the efficiency.

These observations show that we are indeed able to derive clusters of objects together with intensional descriptions in reasonable time; and still with a reasonable degree of detail. Furthermore, 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 would be con-

[6] I. e., for each set of formal concepts, there exists always a unique greatest common subconcept and a unique least common superconcept.

[7] www.navicon.de

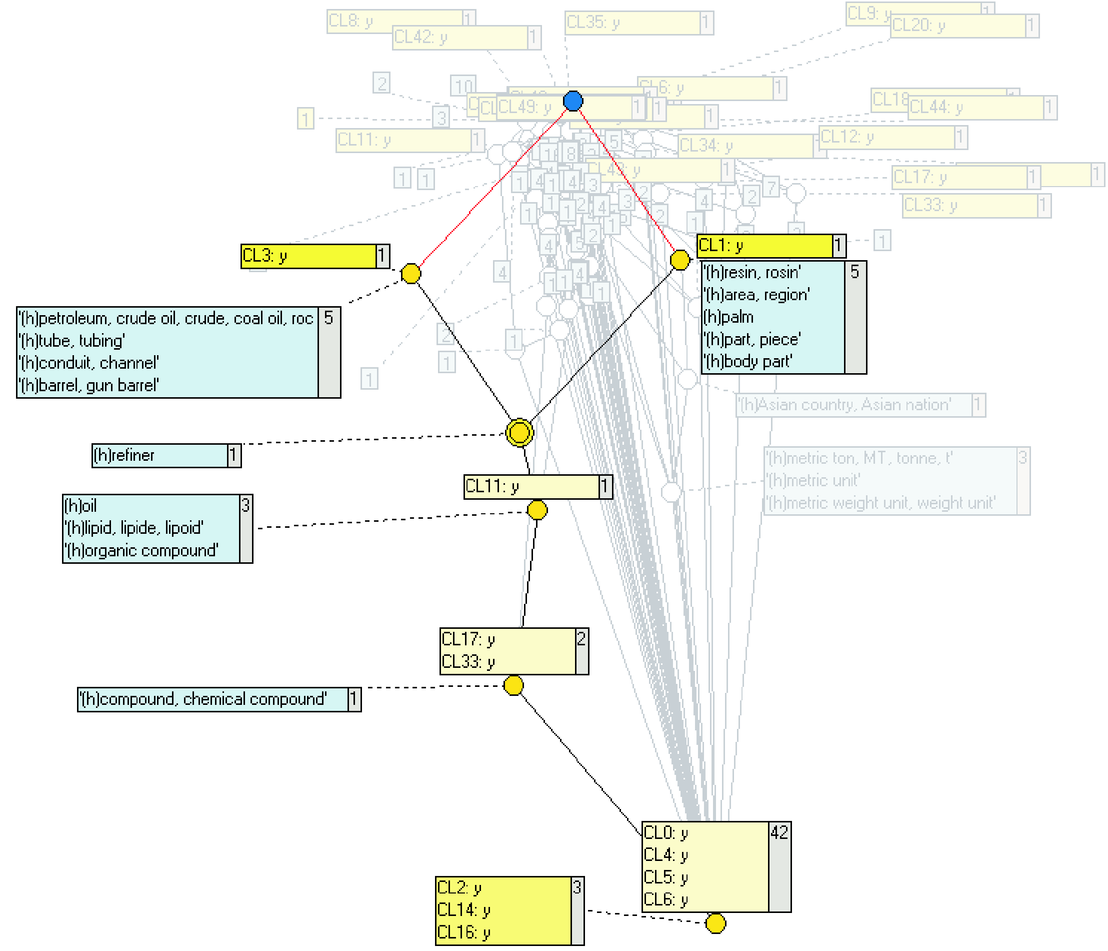

Figure 1 The resulting conceptual clustering of the text clusters (highlighting the concepts related to (oil) refinement).

sidered directly for computing the concept lattice, it could not be guaranteed that the structure of the lattice does not change.

3.3 Analyzing the Document Clusters

Let us show another example of analyzing the documents by our method. In order to give a first hint where to discover interesting structures, we applied first a magnetic spring algorithm for graph visualization[8] for recognizing which clusters are related. A part of the resulting graph is shown in Figure 2. Based on the cosine similarity, it tries to map the clusters into the Euclidean plane such that clusters with similar centroids attract each other, and clusters with different centroids repel each other. Strong similarity (with respect to a given threshold, in our example 75 % of the maximal similarity) is indicated by a line between the clusters. In the diagram, we see for instance that the Clusters 8, 17, 33, 34, 44 and 49 have similar centroids.

The term in parentheses behind a cluster number in the diagram indicates to which Reuters topic the majority of the documents in the cluster were assigned by the Reuters experts. Of course one does not have this additional information when clustering documents in an unsupervised way. We added this information for simplifying the evaluation. In an unsupervised setting, one could display the most important term(s) describing the cluster.

In order to analyze the similarity of the Clusters 8, 17, 33, 34, 44 and 49 conceptually, we restrict the object set of the formal context to just those clusters, and recompute the concept lattice. The result is shown in Figure 3. The lattice provides a lot of details which can be explored interactively using Cernato (as in Fig-

[8] http://java.sun.com/applets/jdk/1.0/demo/GraphLayout/

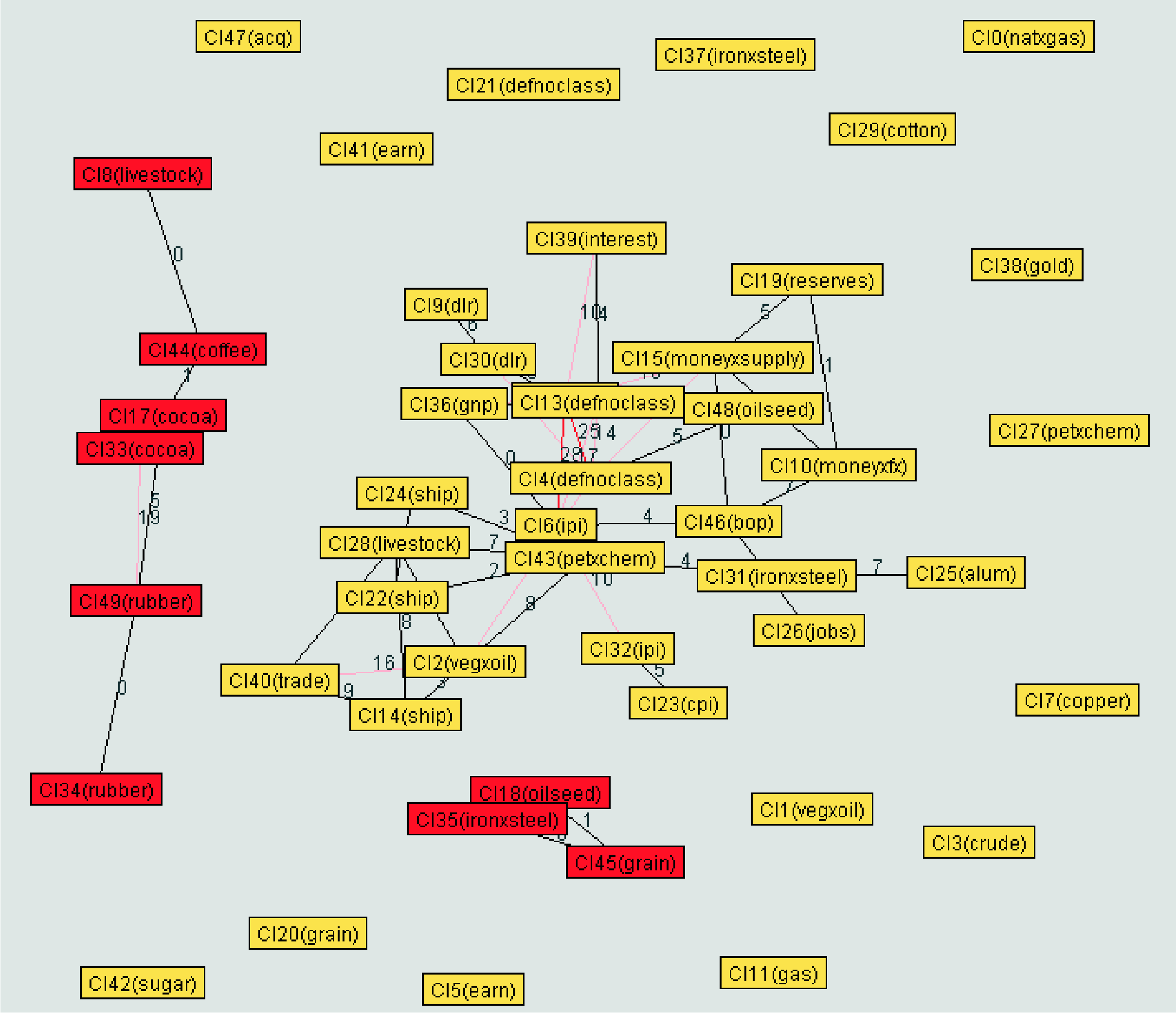

Figure 2 Graph showing (distance-based) similarities between the text clusters

ure 1).

In the diagram, we first observe that the concepts labeled by ‘beverage’ and ‘latex’ partition the set of the six clusters under consideration: The extent of the former is {CL 8, CL 17, CL 33, CL 44}, while the extent of the latter is {CL 33, CL 34, CL 49}. The extent of the ‘beverage’ concept is further split into two disjoint sets: the extent {CL 8, CL 44} of the concept labeled by ‘coffee’, and the extent {CL 17, CL 33} of the concept labeled by ‘cocoa’. Checking this observation with the hand-crafted Reuters topics, one observes that most of the documents contained in these clusters are indeed about coffee and cocoa, resp.

The documents related to latex, on the other side, are grouped together in the extent of the right-most context. As we can see, a part of them (namely the documents contained in cluster 49) is also addressing topics like buffer stocks and international [organizations].[9] In fact, when looking at selected documents, one observes that they address for instance negotiations about the regulation of rubber prices depending on the volume of buffer stocks.

The topics ‘buffer stocks’ and ‘international’ provide also a bridge to the cocoa related documents: All[10] the Reuters documents talking about cocoa also address buffer stock issues, while those contained in Cluster 33 additionally have international organizations as topic.

When checking the concept intent of the concept labeled by ‘CL 8’, one observes a large diversity of topics: pork, . . . , music, coffee, food, beverage. In

[9] The labels ‘non-market economy’, ‘socialism’, etc. are an artifact of our mapping of words to synsets, as ‘International’ was interpreted as noun. We plan to add a part–of–speech tagger to overcome this problem.

[10] Here, ‘All’ means more specifically ‘all, up to the precision reached by BiSec–k–Means’.

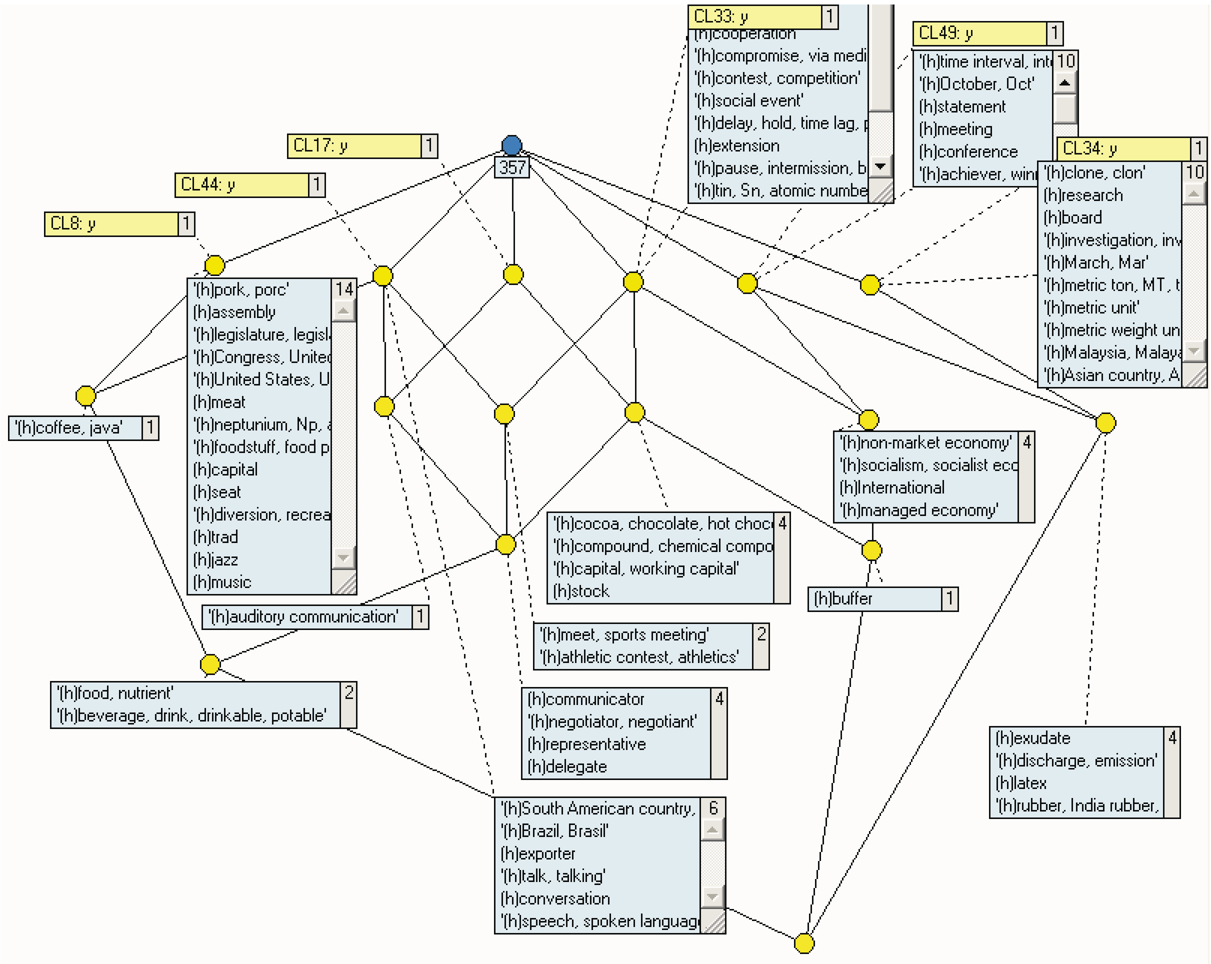

Figure 3 Concept lattice focusing on the clusters 8, 17, 33, 34, 44 and 49.

fact, when reading the documents in Cluster 8, one can observe that many of them are about livestock, and only four of them are about coffee. Thus, in this case, BiSec–k–Means has performed badly, and put together unrelated documents in one cluster. This example shows how one can identify inconsistencies in the results of non-conceptual clustering by using Formal Concept Analysis.

4 Related work

In [13], Pantel and Lin introduce an algorithm called CBC (Clustering by Committee). Committees are disjoint subsets of the object set which are distributed as homogeneous as possible over the object space. Iteratively, documents are assigned to the closest committee, or introduce a new committee if the existing committees are too far away, or are ignored if they are just between existing committees. CBC provides more precise descriptions for the clusters, but does not cover all objects. It could be used instead of BiSec–k–Means in our approach.

In [7], Karypis and Han show that cluster centroids can be used to summarize the content of a cluster. They state that the most important terms in a cluster centroid are the terms with the highest weight. This observation underlies our approach in Section 2.6, where we use only the highly weighted terms to describe the content of the cluster. We differ from their approach in that we additionally make use of WordNet.

Buenaga Rodríguez et. al. [3] and Ureña Lóez et. al. [9] show a successful integration of the WordNet resource for a document categorization task. They use the Reuters corpus for evaluation and improve the classification results of the Rocchio and Widrow-Hoff algorithms by 20 points. In contrast to our approach, they *manually* select synsets for each category and add the terms contained in the synsets with certain weights to the term vectors.

In [6], WordNet is used for word sense disambiguation. Gonzalo et.al. *manually* build a synset vector.

They show in an information retrieval setting the improvement of the disambiguated synset model over the word vector model. In contrast to our approach, they (as well as [3] and [9]) do not make use of WordNet relations other than hypernyms.

Conceptual clustering with Formal Concept Analysis has been discussed in [17, 1, 11, 18]. Another approach to Conceptual Clustering is for instance discussed in [10]. Formal Concept Analysis differs from them in that it does not make use of any heuristics (including arbitrary start settings) and allows for overlapping clusters. Compared to non-conceptual clustering approaches, all conceptual clustering approaches have in common less computational efficiency. Our paper is an approach to overcome this drawback.

5 Conclusion and Future work

In this paper, we discussed a way of combining the efficiency of a common (non-conceptual) clustering technique with the intensional descriptions provided by a conceptual clustering approach. We showed how this approach can be applied to a corpus of documents. We first clustered the documents using BiSec–k–Means, using a thesaurus as background knowledge. This clustering reduces the number of documents, so that they can be clustered conceptually in the second step using Formal Concept Analysis.

As the work presented here is a first attempt to combine conceptual and non-conceptual clustering, many interesting research topics remain. For instance, it seems promising to check if non-disjoint clustering techniques (instead of BiSec–k–Means) might give better results together with FCA for describing the documents, since FCA explicitly allows a one–to–many assignment between objects and attributes. We will also study alternatives to *tfidf* and cosine for measuring similarity.

Another important question is how the resulting concept lattice shall be presented to the user. We will check which approaches are best fit to this purpose. One way is for instance to use Cernato as we did for this paper. Another way is to derive conceptual scales [4] by grouping together the most related sets of terms. These conceptual scales can then be visualized using TOSCANA [8, 19, 12]. The resulting concept lattice may also be accessed based on iceberg concept lattices [18], or as discussed in [1] or [2]. We will test these approaches also on domain-specific ontologies other than WordNet.

Interesting from a structural point of view is how the tree structure from the non-conceptual clustering by BiSec–k–Means fits with the concept lattice. If it is (more or less) embedded in the concept lattice, this fact can be exploited for navigation and retrieval tasks.

Another interesting question is if Formal Concept Analysis can be used for automatically computing intensional descriptions of the clusters generated by the non-conceptual clustering algorithm. These descriptions will consist of conjunctions of terms. It has to be defined what a ‘globally optimal’ description for the clusters is. Then an algorithm for computing such a description has to be developed (compare also with [10]).

From our experience with the application described in this paper we believe that it is promising to combine the advantages of an intensional description of conceptual clustering with the efficiency of non-conceptual clustering. But further work has to been done to bring this combination to its full potential.

References

1. C. Carpineto and G. Romano. GALOIS: An order-theoretic approach to conceptual clustering. In *Machine Learning, Proc. ICML 1993*, pages 33–40. Morgan Kaufmann Publishers, 1993.
2. R. Cole and G. Stumme. CEM – a conceptual email manager. In B. Ganter and G. W. Mineau, editors, *Conceptual Structures: Logical, Linguistic, and Computational Issues. Proc. ICCS ’00*, volume 1867 of *LNAI*, pages 438–452, Heidelberg, 2000. Springer.
3. Manuel de Buenaga Rodríguez, José María Gómez-Hidalgo, and Belén Díaz-Agudo. Using wordnet to complement training information in text categorization. In N. Nicolov and R. Mitkov, editors, *Recent Advances in Natural Language Processing II: Selected Papers from RANLP ’97*, Amsterdam-Philadelphia, 2000. John Benjamins.
4. B. Ganter and R. Wille. Conceptual scaling. In F.Roberts, editor, *Applications of combinatorics and graph theory to the biological and social sciences*, pages 139–167, New York, 1989. Springer.
5. B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999.
6. J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarrán. Indexing with wordnet synsets can improve text retrieval. In *Proceedings ACL/COLING Workshop on Usage of WordNet for Natural Language Processing*, 1998.
7. George Karypis and Eui-Hong Han. Fast supervised dimensionality reduction algorithm with applications to document categorization and retrieval. In Arvin Agah, Jamie Callan, and Elke Rundensteiner, editors, *Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management*, pages 12–19, McLean, US, 2000. ACM Press, New York, US.
8. W. Kollewe, M. Skorsky, F. Vogt, and R. Wille. TOSCANA – ein Werkzeug zur begrifflichen Analyse und Erkundung von Daten. pages 267–288, 1994.
9. J. M. Gómez Hidalgo L. A. Ureña Lóez, M. de Buenaga Rodríguez. Integrating linguistic resources in

tc through wsd. *Computers and the Humanities*, 35(2):215–230, 2001.

10. R. S. Michalski and R. Stepp. Learning from observation: Conceptual clustering. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning, An Artificial Intelligence Approach*, volume II, pages 331–363, Palo Alto, 1983. TIOGA Publishing Co.
11. G. Mineau and R. Godin. Automatic structuring of knowledge bases by conceptual clustering. *IEEE Transactions on Knowledge and Data Engineering*, 7(5):824–829, 1995.
12. The ToscanaJ Project: An Open-Source Reimplementation of TOSCANA.
    http://toscanaj.sourceforge.net.
13. Patrick Pantel and Dekang Lin. Document clustering with committees. In *Proceedings of SIGIR02, Tampere, Finland*, 2002.
14. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
15. G. Salton. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1989.
16. M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.
17. S. Strahringer and R. Wille. Conceptual clustering via convex-ordinal structures. In O. Opitz, B. Lausen, and R. Klar, editors, *Information and Classification*, pages 85–98, Berlin-Heidelberg, 1993. Springer.
18. G. Stumme, R. Taouil, Y. Bastide, N. Pasqier, and L. Lakhal. Computing iceberg concept lattices with Titanic. *J. on Knowledge and Data Engineering*, 42:189–222, 2002.
19. F. Vogt and R. Wille. TOSCANA – a graphical tool for analyzing and exploring data. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing ’94*, volume LNCS 894, pages 226–233, Heidelberg, 1995. Springer.