# Extracting Domain Ontologies with CORDER

Technical Report kmi-05-14
September 2005

Camilo Thorne, Jianhan Zhu, Victoria Uren

# EXTRACTING DOMAIN ONTOLOGIES WITH CORDER

CAMILO THORNE*, JIANHAN ZHU†, AND VICTORIA S. UREN†

Abstract. The CORDER web mining engine developed by the Knowledge Media Institute[1] and Stella Group[2] computes a lexical co-occurrence network out of websites – a binary relation R. A natural extension of CORDER would be that of learning an ontology. However, our work shows that co-occurrence proves insufficient to discover concepts and conceptual taxonomies (i.e. very simple ontologies) out of this network. To tackle this problem two unsupervised learning methods were studied based, on the one hand, on set similarity (and thus on a set-based representation of the data) and, on the other hand, on cosine similarity (and thus on a vector-space representation of the data). The underlying idea being that of taking into account, for the clustering, as features, their related co-occurring entities (and thus the indirect links among the entities), as suggested by O. Ferret (cf. [4]). For the purposes of this study, we restricted ourselves to (solely) research areas. The most promising results in our experiments were given by the vector-space representation. To validate the results we used the ACM classification of computer science research areas as our gold standard.

## Contents

*Date:* KMi – Open University, September 2005.

* camilo.thorne@laposte.net
†{v.s.uren, j.zhu}open.ac.uk

[1] http://kmi.open.ac.uk

[2] http://www.stela.org.br/

## 1. Motivation

1.1. **Extending CORDER.** The CORDER web-mining algorithm computes a relation that we will denote $R$ holding between named entities of different types (research areas, researchers, organizations, etc.), according to relation strength measurements based on coocurrence, and that are recognized and extracted by the NER platform (i.e. a shallow parser) ESPOTTER (cf. [3], [2]). This relation is computed by way of a proximity graph (a lexical co-ocurrence network). And the associated similarity/distance matrix [1] (a symmetric square matrix of real numbers of order $n \times n$) can be afterwards used to index or rank the entities, since the graph can be embedded into a $n$-dimensional vector space –where $n$ denotes the size of the data set extracted from the Southampton Department of Computer Science website (i.e. the set of named entities extracted from it). The open problem left is whether this output, when we restrict $R$ to entities of the same type, can serve to learn on, again, an unsupervised basis, a domain ontology (a taxonomy to be more precise) describing the domain of artificial intelligence.

1.2. **The methods for extracting a taxonomy of research areas.** The main aim of this work is therefore to try to extract a hierarchy of research areas (an ontology restricted to taxonomical relations) to a certain degree by means of hierarchical agglomerative clustering (using Ward's well-known algorithm, cf. [5]. [1] and [12]). Now, clustering (either hierarchical or partitional) works on some distance/similarity previosuly defined measure among the clustered objects (and a previously computed distance/similarity measure). We thus experimented with three different clustering methods based on three different metrics, namely:

(1) A variant of the well-known set similarity measure.
(2) The cosine similarity measure.
(3) CORDER's own distance measure.

Results were then validated by comparing them to the ACM (Association of Computer Machinery) classification – our gold standard.

1.3. **The reasons behind hierarchical clustering.** Learning an ontology (a taxonomy) can be understood as the process of inductively inferring a hierarchy of concepts from a collection of individuals holding properties and relations among them, a dataset – the real-world domain. This involves three main stages:

(1) To achieve this conceptual structure we need some kind of intermediate representation (of some lower level), as required by clustering algorithms (such as Ward's, cf. [1], [12] and [5]).
(2) Furthermore, it has been argued (by O. Ferret) that (from the standpoint of lexical and textual semantics) concepts (intensions) can be taken to be collections of (named) entities: an equivalence class modulo synonymy – clusters. And that the clustering features are nothing but cooccurring entities under the assumption that shared meaning among entities involves shared coocurrence with other entities (cf. [4] – and [9] for a more general discussion on ontologies and concept taxonomies).
(3) Therefore, to discover concepts and concept subsumption (yielding a taxonomy) we need a method capable of, on the one hand, clustering the entities w.r.t. their meanings, yielding concepts, and to hierarchically structuring

[1] Similarity and distance are dual notions.

these clusters. Whence the utility of hierarchical clustering to discover this conceptual structure (cf. [10]).

1.4. **Modelling the data.** The objects or individuals to be clustered need therefore to be previously modelled in order to yield concepts by way of clusters. We thus chose two different representations of the data, namely: **(1)** as sets of features and **(2)** as feature vectors – and defined two metrics based, respectively on set distance and cosine distance, to build a distance matrix so as to perform clustering. These representations of the data are quite natural, since entities are structured by CORDER (and likewise by any other algorithm building a cooccurrence matrix) into a graph. Their features are then nothing but their immediate neighbourhood in the graph. Clustering is thus performed on a set or vector-space representation of these subgraphs.

1.5. **Research areas as spanning trees.** CORDER computes, as already noticed, a lexical coocurrence network. Concentrating on any one of its vertex and its immediate neighbourhood yields an acyclic connected component – i.e. a (weighted) tree, whose root can be seen as the research area, the individual, and whose leaves as its features or properties. Moreover, leaves can be ordered following their edge's weight from lightest to heaviest. In the case we are studying, the objects to be clustered are research area entities and the features of a research area nothing but its related entities of the same type – i.e. the related research areas of its neighbourhood, ranked according to the aforementioned order. As a matter of fact, a subset of this neighbourhood is enough, since many of these research areas can be seen as irrelevant – their relation strength might be too low. We assume throughout this paper that only a limited number (the top 10) are enough, because of this reason.

1.6. **A word on the weighing and on the notation.** We may further assume that it is the top three or so the so to speak critical classification features – due to the fact that, statistically, they tend to outweigh the others and define a weighting (based on a monotone increasing sequence of numbers – the Fibonacci sequence), based on this trend of the data. On what follows we will denote $\alpha$ a research area (or area) and $A$ their set. While we will denote $a$ a feature of a reasearch area $\alpha$, and $P$ their set. For a more detailed account on the metrics from which ours were adapted cf. [5] and [12].

## 2. Set distance clustering method

2.1. **Overview.** In this section we will represent a research area $\alpha$ as an ordered set $s_\alpha = \{a_1, ..., a_n\}$ of features, sorted w.r.t. relation strength and thus to cooccurrence – its features being its nearest research areas (w.r.t., again, CORDER). We denote their set by $D$. $D$ will thus denote our input dataset. Since arguably the most relevant features of a research areas are its best 10, we limit ourselves to research areas of that size. Moreover, we assume as an hypothesis that the more their strength, the more their relevance for the clustering, and define a weigthing over them based on this intuition – clustering should be possible with, say, the best 3 (of the 10), if it so happens. The weighting is due to the fact that plain set similarity [2] is too coarse a similarity measure and unadapted to sets upon which an order may have been defined. This weighting is achieved by using the Fibonacci function (see its graph on Figure 1), giving way to the strength or weight function $w_\alpha$ particular to each area and defined as the $(i-1)$-est Fibonacci number for the $i$-est research area (from worst to better). By doing this, we can ascribe a weight of 1 or 2 to the worst three and 55 to the best – this area being *ex hypothesi* quite determinant. The $\Delta$ distance/similarity function among research areas then turns out to be some ratio of their shared features to their overall features. Pairwise similarity/distance is stored in a similarity/distance matrix.

### 2.2. **Defining the metric.**

In this section we formalize the notions discussed above and state their basic properties:

**Definition 2.1. (Weight)** *Let $\alpha \in A$, and let $(a_i)_{i\in\{1,\ldots,n\}}$ be an enumeration of its features (ordered w.r.t. relation strength). The* weight *function for $\alpha$'s features is the function $w_\alpha : (a_i)_{i\in\{1,\ldots,n\}} \to \mathbb{N}$ such that:*

$$w_\alpha(a_i) = fib(i-1).$$

That is, we map the $i$-est related area of $a$ onto the $(i-1)$-est term of the sequence of the fibonacci numbers (see again Figure 1). This allows for their being sufficiently scattered (just think in the graph of the Fibonacci function). Moreover, this gives a far greater weight to the most relevant areas, and adds to the intuition that it is the most relevant one the one essential for the classification. As said previously we will only consider sequences of research areas of length $n = 10$, i.e.:

$$\begin{gathered} w_\alpha(a_1) = 1, \\ \vdots \\ w_\alpha(a_{10}) = 55. \end{gathered}$$

Remark that there is an $w_\alpha$ function for each area $\alpha$ (we have thus defined a whole family). The next step is to define the similarity/distance measure:

**Definition 2.2. (Distance)** *Given $s_\alpha, s_{\alpha'} \in D$ we define their* distance *as the the function $\Delta : D \times D \to \mathbb{R}$ such that:*

$$\Delta(s_\alpha, s_{\alpha'}) = 1 - \left( \frac{\sum\limits_{a \in s_\alpha \cap s_{\alpha'}} w_{s_\alpha}(a) + w_{s_{\alpha'}}(a)}{\sum\limits_{a \in s_\alpha \cup s_{\alpha'}} w_\alpha(a) + w_{\alpha'}(a)} \right).$$

The following immediate properties are quite important:

**Proposition 2.1.** *We have that, for any $s_\alpha, s_{\alpha'}, s_{\alpha''} \in D$ :*

- $\Delta(s_\alpha, s_{\alpha'}) = 0$ *(null distance).*
- $\Delta(s_\alpha, s_{\alpha'}) = \Delta(s_{\alpha'}, s_\alpha)$ *(symmetry).*
- $\Delta(s_\alpha, s_{\alpha'}) \leq \Delta(s_\alpha, s_{\alpha''}) + \Delta(s_{\alpha''}, s_{\alpha'})$ *(triangularity).*

Whence:

[2] Let $E$ be a set. Given $A, A' \subseteq E$ we define their *set similarity* as the function $sim : \wp(E) \times \wp(E) \to \mathbb{R}$ such that:

$$sim(A, A') = \frac{\sharp(A \cap A')}{\sharp(A \cup A')}.$$

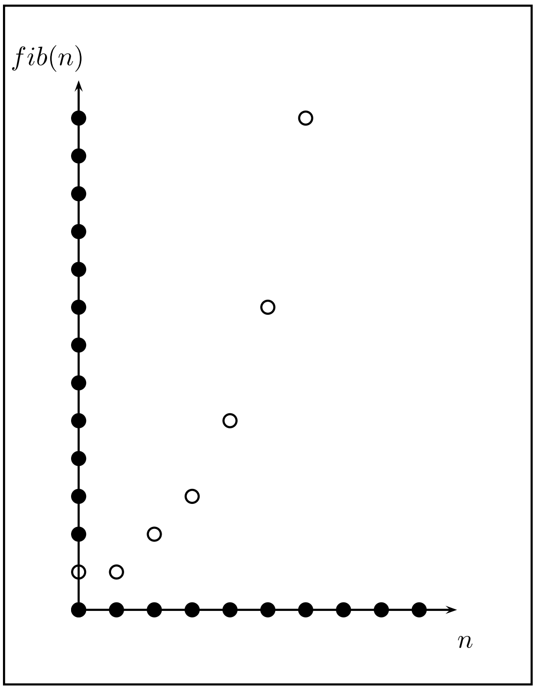


FIGURE 1. Graph of the Fibonacci function.

**Corollary 2.1.** *$\Delta$ is metric and $(D, \Delta)$ a metric space.*

In other words, this re-definition of set distance is mathematically sound. It remains to be seen how well it captures semantic similarity.

## 3. Cosine distance clustering method

3.1. **Overview.** We define in this section another distance measure, which is too a metric, based on or inspired by the cosine similarity measure [3] for vectors in the $m$-dimensional $\Re^m$ euclidian vector space [4] (cf. [12]). In order to do this, we define (on basis of the previous one) a new weigthing, and map each research area onto a normalized vector of length $m \leq (10 * n)$ – the dimension of the term space we thereby build (since for each of the $n$ reseach areas, we have 10 features). Thus an area $\alpha$ is now associated to (and represented by) a vector $\overrightarrow{x_\alpha}$ of $\Re^m$. Pairwise similarity/distance is again stored in a similarity/distance matrix. The main aim of introducing this rather well known metric is that of providing a basis of comparison with the previous one. It actually proved to yield better results (qualitatively speaking) than the other two.

[3] Let $\overrightarrow{x}$, $\overrightarrow{y}$ be two vectors of the normed space $\Re^m$. Then their *cosinus similarity* is the function $sim : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ such that:

$$sim(\overrightarrow{x}, \overrightarrow{y}) = \frac{\overrightarrow{x} * \overrightarrow{y}}{\|\overrightarrow{x}\| * \|\overrightarrow{y}\|}.$$

[4] i.e. a normed vector space provided with a inner product.

### 3.2. Defining the metric.

**Definition 3.1. (Weight)** *Let $a \in N$ and $\alpha \in A$. The* weight *of $a$ relatively to $\alpha$ is given by the function $w'_\alpha : N \to \mathbb{R}$ such that*

$$w'_\alpha(a) = \begin{cases} w_\alpha(a) \text{ if } a \text{ is a feature of } \alpha \\ 0 \text{ otherwise.} \end{cases}$$

So, formally, we represent each area $\alpha$ by a vector

$$\overrightarrow{x_\alpha} = (w'_\alpha(a_1), ..., w'_\alpha(a_m))^T$$

of weights in $\mathfrak{R}^m$. We denote the corresponding subspace of $\mathfrak{R}^m$ they engender by $\mathfrak{D}^m$. $\mathfrak{D}^m$ is thus (so to speak) our input dataset.

**Definition 3.2. (Cosinus Distance)** *Given $\overrightarrow{x_\alpha}, \overrightarrow{x_{\alpha'}}$ in $\mathfrak{R}^m$ we define their* cosinus distance *as the the function $\Delta' : \mathbb{D}^m \times \mathbb{D}^m \to \mathbb{R}$ such that:*

$$\Delta(\overrightarrow{x_\alpha}, \overrightarrow{x_{\alpha'}}) = 1 - \left( \frac{\sum_{i=0}^{m} [w'_\alpha(a_i) * w'_{\alpha'}(a_i)]}{\sqrt{\sum_{i=0}^{m} [w'_\alpha(a_i)]^2} * \sqrt{\sum_{i=0}^{m} [w'_{\alpha'}(a_i)]^2}} \right).$$

Obviously, $\Delta'$ is also metric, and $\mathfrak{D}^m$ a metric space.

## 4. CORDER's relation strength – the direct link clustering method

As already mentioned, CORDER's relation strength measure can be also taken to be a metric $\Delta''$ – over the set of named entities and in particular over $A$, the set of reseach area entities. As opposed to the former metrics, the features taken into account are statistical measures drawn from the coocurrence frequency of any two entities throughout the corpus (the Southampton website). See [2] for further details on how this measure is actually defined. The dataset used in this case was considerably bigger (343 entities) even though it encompassed the one that can be seen on Figures 5 and 6 (of 30). Figure 7 shows a partial snapshot of the results obtained (cut at a distance of 0.5). The results look like a collection of disjoint clusters mainly because, on the one hand, CORDER's proximity matrix is very sparse (meaning that many entities may not cooccur at all), and on the other hand due to the threshold. We thus obtain a forest that is subsequently flattened.

## 5. Ward's algorithm and clustering

In order to infer or learn the ontology, we used Ward's agglomerative hierarchical clustering algorithm using the single linkage criterion (cf. [5], [1]), which is to say, current distance minima. This algorithm builds the clusters bottom-up – i.e. it begins by building a cluster out of each of the elements of a dataset $X = \{x_1, ..., x_n\}$, and proceeds then to iteratively merge them pairwise (unless a threshold for minima is chosen). A cluster $c$ being a certain subset of $X$. It relies upon defining a distance measure upon clusters. Let $\delta$ denote the distance function defined on data inputs. For single linkage clustering, the inter-cluster distance is the function $\delta' : \wp(X) \times \wp(X) \to \mathbb{R}$ such that:

$$\delta'(c, c') = \min_{x \in c, x' \in c'} \delta(x, x').$$

In other words, the distance between two clusters is the minimum distance among the objects they contain. In the case we are studying $\delta$ is one of the previously defined distance measures – i.e. $\delta \in \{\Delta, \Delta', \Delta''\}$.

**Algorithm 1** Ward's Algorithm (Single-Link Criterion)

```
procedure $HIERACHICAL(\{x_1, ....x_n\})$
    for $1 \leq i \leq n$ do
        $c_i \leftarrow \{x_i\}$;
    end for
    $C \leftarrow \{c_1, ...., c_n\}$;
    $C' \leftarrow C$;
    while $\sharp(C) \geq 1$ do
        $(c, c') \leftarrow \arg \min_{(c,c') \in C \times C} \delta'(c, c')$;
        $c'' \leftarrow c \cup c'$;
        $C \leftarrow (C - \{c, c'\}) \cup \{c''\}$;
        $C' \leftarrow C' \cup \{c''\}$;
    end while
    return $C'$;
end procedure
```

An algorithm of (worst case) complexity $O(n^3)$. There are $n$ merging steps (iterations of the while loop), and at each step $O(n^2)$ comparisons to find the minima, $n$ being the size of $X$. From the standpoint of graph theory, the algorithm can be seen as computing the least connected components of a spanning tree (the graph represented by the distance matrix, which is an adjacency matrix too).

## 6. Results and further work

### 6.1. The validation method.

6.1.1. *A word on the implementation.* We implemented the distance measures and the clustering algorithm (together with the needed data types and structures, such as sets and clusters) in Java. The input dataset with which we worked being retrieved from a (small) MSAccess database (through the JDBC API) comprising 30 research areas (see Figure 5 and 6). Outputs were stored as .txt files. To visualize the resulting dendogram we made use of the **Clustan Graphics**$^{\mathbf{TM}}$ tool. The output dendogram is a binary tree, but it can be turned easily into an $n$-ary tree by thresholding cluster distance (thereby reducing its size). We plan to implement a graphical visualization of the output in the near future. Direct link clustering was performed by extending CORDER and operating on a dataset of 343 named entities (see Figure 7 for a snapshot).

6.1.2. *The gold standard.* The clustering results were validated by comparing them to the Association of Computer Machinery (ACM) [5] classification (which can be assumed to be an ontology), which was chosen as our gold standard. ACM classification was chosen because of its being the most comprehensive one, although it is quite old (it has not been greatly updated since 1998 and its basic structure is

[5] Found at http://www.acm.org/class/1998.

even older) and as such lacking in many new research areas that have come into existence since (like the semantic web, to mention its most notorious gap).

6.1.3. *Adapting the precision and recall IR measures.* In order to better evaluate our results we chose to adapt the well known precision and recall information retrieval measures (cf. [12] for more details on the standard definition for text mining and document indexing systems) to the present case. Loosely speaking, precision will be understood as the ratio of the number of relevant clustered areas belonging to a target set to the total number of areas, whereas recall as the ratio of the number of these relevant clustered areas to the number of expected clustered areas. In computing recall and precision we chose as well to put aside the structure of the results obtained, flattening the trees. The target set is, broadly speaking, the artificial intelligence domain (as described by the ACM classification). Clustered areas are understood as the entities belonging to a distinguished cluster of the learned clusters, namely one containing the artificial intelligence area. The relevant areas, are, naturally, the areas in this distinguished cluster that belong to the target set. The expected clustered areas are the areas in the dataset that belong to the target set, whether or not contained by this distinguished cluster – but that we would expect clustered together with artificial intelligence. Formally:

**Definition 6.1. (Precision and recall)** *Let $DS \subseteq A$ be an input dataset of research areas, $C$ the set of clusters computed by Ward's algorithm and $c_{AI} \in C$ some cluster containing the artificial intelligence research area. Then* precision *and* recall *are the quantities:*

$$PR = \frac{\sharp\{\alpha \in c_{AI} | \alpha \textit{ belongs to the ACM AI subtree }\}}{\sharp\{\alpha \in DS | \alpha \in c_{AI}\}}.$$

$$RE = \frac{\sharp\{\alpha \in c_{AI} | \alpha \textit{ belongs to the ACM AI subtree }\}}{\sharp\{\alpha \in DS | \alpha \textit{ belongs to the ACM AI subtree }\}}.$$

6.2. **Results.**

6.2.1. *Cosine and set distance.* The results obtained were better when clustering with $\Delta'$, cosine distance (Figure 4), than with $\Delta$, set distance (Figure 3), although the dendograms may look quite similar at first sight. We have underlined in grey the research areas globally clustered along with artificial intelligence. Moving to the root of the subtree of which they are the leaves we obtain a representation of (roughly) the artificial intelligence domain. The cosine distance dendogram clusters with artificial intelligence areas such as robotics or agents which do belong to it in the ACM classification. While in the set distance dendogram we can see it includes areas such as concurrency – which belong to the domains of electronics and software engineering, the other main areas of expertise of Southampton's computer scientists. Regarding $\Delta''$, CORDER distance, since the clustering method employed was different, a direct comparison is not possible. However, recall and precision measures can be used to tackle this.

6.2.2. *CORDER direct link metric.* CORDER's co-ocurrence distance proved to yield, naturally, very different results – the dataset having been far bigger. However, w.r.t. to our gold standard, they proved deceiving, clustering together artificial intelligence and networking (see Figure 7), instead of, for instance, machine learning. This is due to the fact that, as said, the proximity matrix is very sparse,

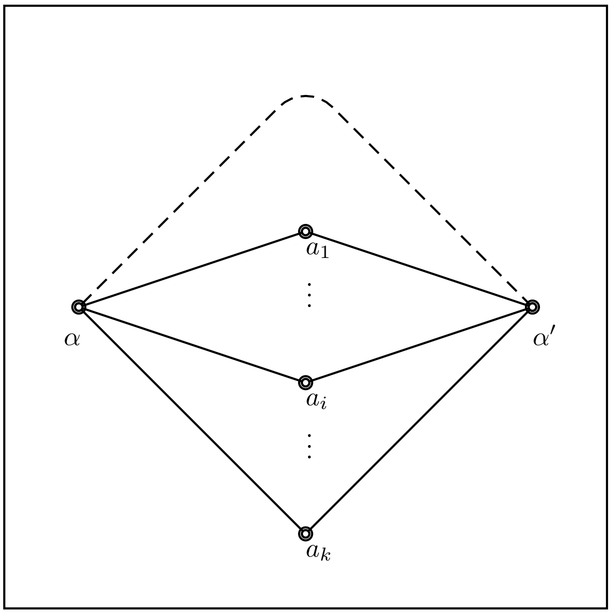


Figure 2. Direct links and indirect links for two areas $\alpha$ and $\alpha'$. The dashed line represents the direct link. The number $k$ of indirect links may be null.

or that the (direct) links between two areas $\alpha$ and $\alpha'$ may have a very low relation strength. The other two methods (using cosine and set distance) yield better results because they use too as features the paths, so to speak, going from $\alpha$ to $\alpha'$ (see Figure 2) – i.e. their being 1-accessible, together with the direct link if relevant enough.

6.2.3. *Precision and recall.* For cosine and set distance we cut the resulting dendograms in a such a way as to obtain a reasonable good cluster or hierarchy of clusters containing the artificial intelligence area, focusing thereafter on the root cluster of this subtree and choosing it as our $c_{AI}$ cluster – see, again, the subtrees from Figures 4 and 3 marked in grey. For the direct link method, the $c_{AI}$ cluster is cluster 24 (see Figure 7). We denote $PR_{set}$, $PR_{cos}$ and $PR_{cor}$ the precision (and similarly for recall) of the set distance, cosine and direct link methods. Whence:

- $PR_{set} = 43.48\%$, $RE_{set} = 57.14\%$.
- $PR_{cos} = 69.23\%$, $RE_{cos} = 64.29\%$.
- $PR_{cor} = 25\%$, $RE_{cor} = 2.86\%$.

Which further argues in favour of our considering $\Delta'$ as the best metric.

### 6.3. **Further work.**

6.3.1. *Clustering the spanning trees.* One of the drawbacks of the method followed (i.e. hirarchical aglommerative clustering) is its sensitivity to irrelevant features, yielding wrong classifications. Since the basic idea is to discover semantic relations by clustering together subgraphs of the lexical cooccurrence matrix computed by CORDER, better results could be attained with more information about the objects of the input dataset, i.e. taking into account related entities of any type whatsoever (research areas, organizations, researchers, etc.) whether all of them or just a subset, as suggested by Ferret in [4]. CORDER distances could be, moreover, taken as a weighting for these features, and the cosine distance as the metric for their associated vector representation.

6.3.2. *Internal and external quantitave validity criteria.* Another issue that remains to be addressed is the study and implementation of quantitative clustering validity criteria – so as to, for example, be able to find the best $n$-ary tree. Since there are only at most $m$ of these trees (for a dataset of size $m$), this problem can be easily adressed as an optimization problem maximizing some cluster validity internal criterion such as Dunn's (i.e. maximizing intra-cluster distance and minimizing intra-cluster distance), as suggested by Bezdek and Pal in [8]. Another reasonable method of accomplishing this would be that of storing the clustering results as an .xml file and then computing its similarity to the XML-formatted version of the ACM classification – a method based in a variant of the well-known Levehnstein edit metric but adapted to XML (see Kovacs et al. in [11] for more details).

6.3.3. *Tagging the internal nodes of the taxonomical tree.* Tagging the internal nodes remains too an open problem, since the dendogram or the metrics give few clues, as they stand, about conceptual subsumption. But that can be accomplished by means of Hearst patterns (cf. [7]).

## References

[1] M. N. Murty A. K. Jain and P. J. Flynn. Data clustering: Review. *ACM Computing Surveys*, 31(3), 1999.

[2] JianhanZhu Enrico Motta Alexandre Gonçalves, Victoria Uren and Roberto Pacheco. Mining web data for competency management. In *Proc. of Web Intelligence Conference 2005 (WI05)*, pages 94–100. IEEE Computer Society, 2005.

[3] Victoria Uren Alexandre Gonçalves and Jianhan Zhu. Adaptive name entity recognition for social network analysis and dommain ontology maintenance. http://kmi.open.ac.uk/publications/papers/kmi.pdf, 2004.

[4] Olivier Ferret. Découvrir des sens des mots à partir d'un réseau de coocurrences lexicales. http://www.lpl.univ-aix.fr/jep-taln04/proceed/actes/taln2004-Fez/Ferret.pdf, 2004.

[5] Menahem Friedman and Abraham Kandel. *Introduction to Pattern Recognition*. World Scientific, 1999.

[6] Nicola Guarino. Formal ontology, conceptual analysis and knowledge representation. http://www.loa-cnr.it/Papers/FormOntKR.pdf, 1995.

[7] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. http://www.cs.utah.edu/classes/cs6936/papers/hearst-coling92.pdf, 1992.

[8] Nickil R. Pal James Bezdec. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3), 1998.

[9] Daniel Kayser. *La représentation des connaissances*. Hermes, 1997.

[10] Latifur Khan and Feng Luo. Hierarchical clustering for complex data. *International Journal on Artificial Intelligence Tools*, 14(5), 2005.

[11] Tibor Repasi Lazlo Kovacs and Erika Baksa-Varga. Nearest neighbour search for xml trees. http://www.bmf.hu/conferences/sisy2004/kovacs.pdf, 2004.

[12] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing (Ch.IV)*. The MIT Press, 2002.

[13] Yannis Batistakis Maria Halkidi and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2/3), 2001.

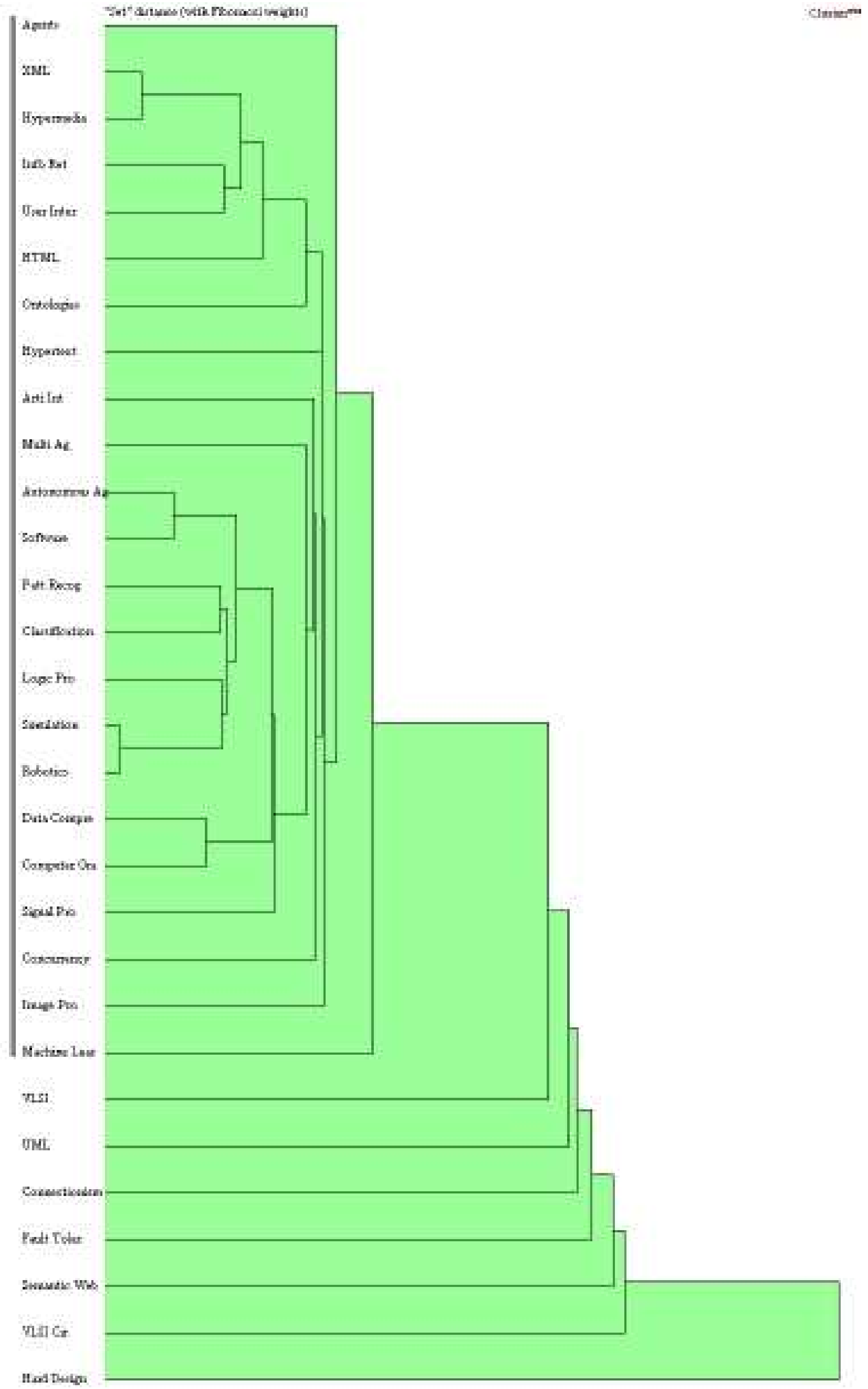


FIGURE 3. Dendogram obtained with set distance. The areas in grey mamrk roughly the artificial intelligence domain. Research areas have been abridged. See the dataset (cf. Figures 5 and 6).

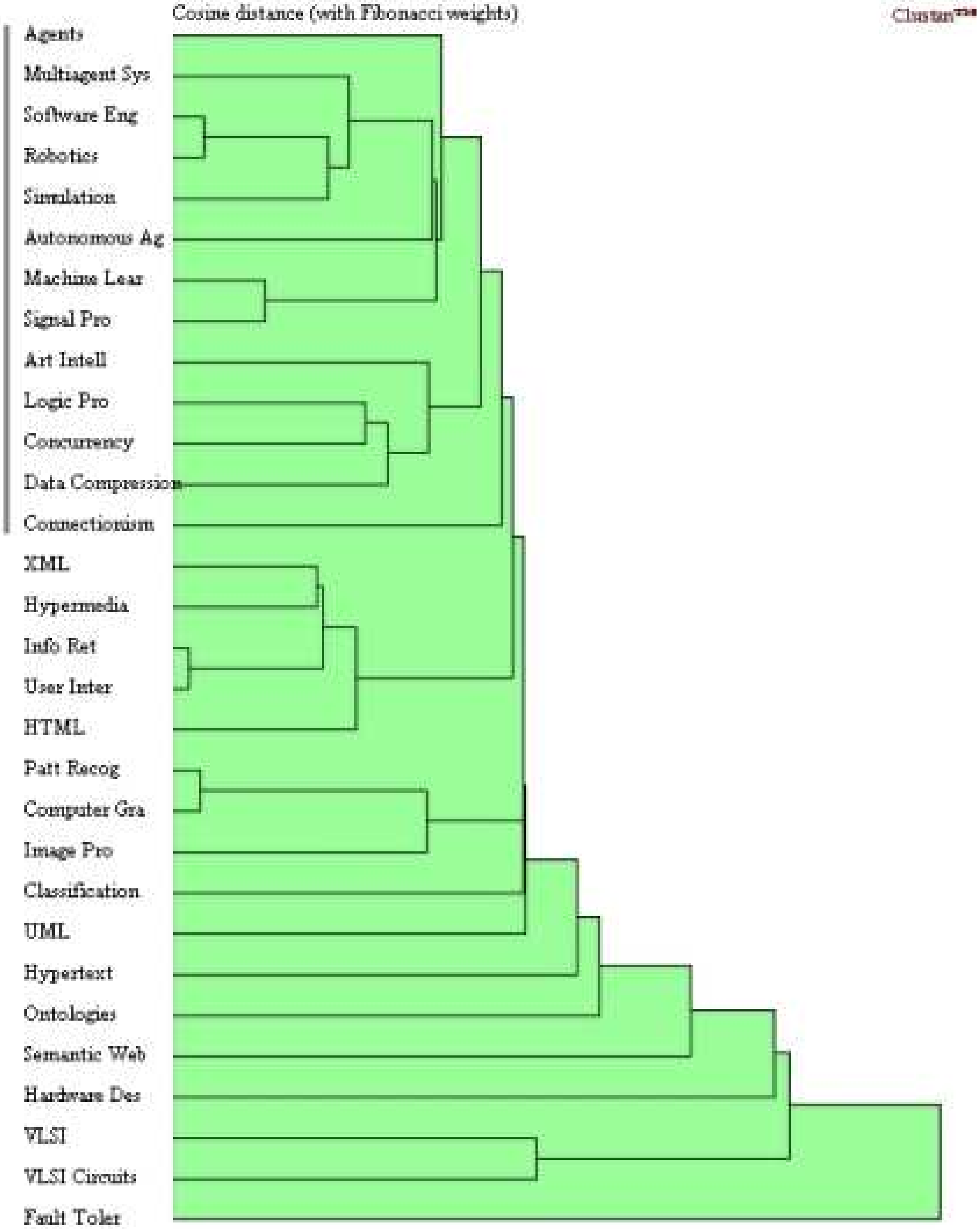
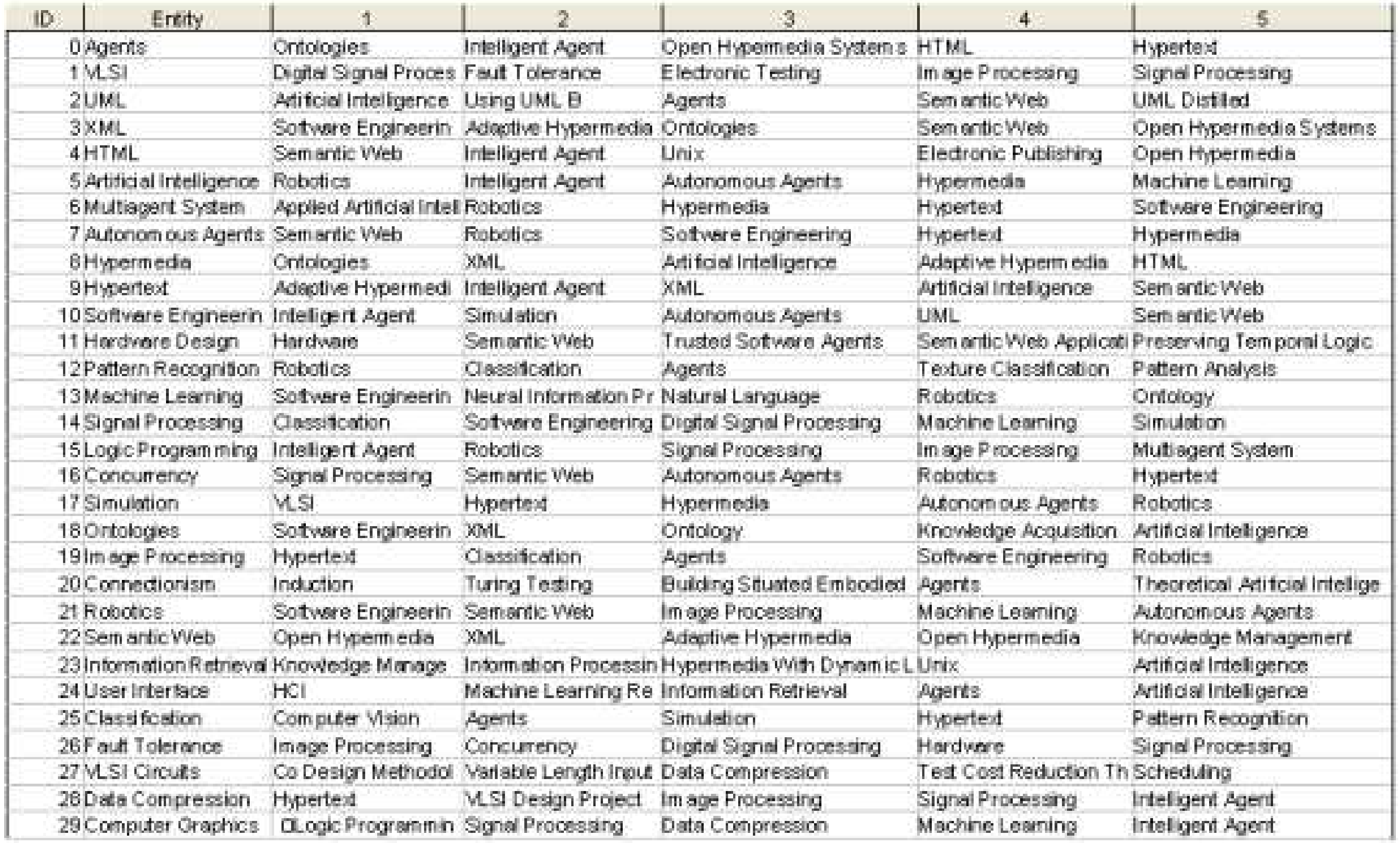
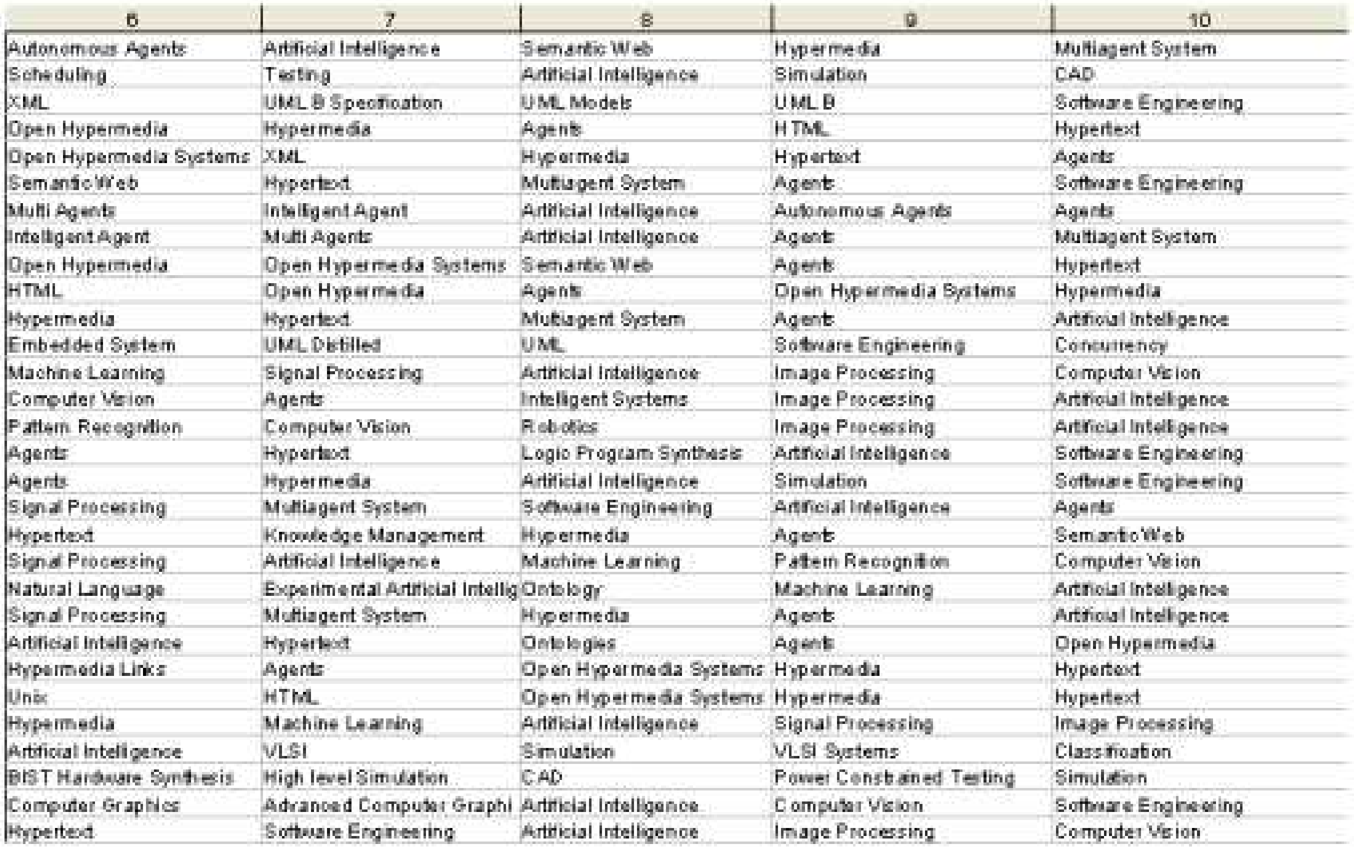


FIGURE 4. Dendogram obtained with cosinus distance. The areas in grey mark roughly the artificial intelligence domain. They are also abridged.



| ID | Entity | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | Agents | Ontologies | Intelligent Agent | Open Hypermedia Systems | HTML | Hypertext |
| 1 | VLSI | Digital Signal Proces | Fault Tolerance | Electronic Testing | Image Processing | Signal Processing |
| 2 | UML | Artificial Intelligence | Using UML B | Agents | Semantic Web | UML Distilled |
| 3 | XML | Software Engineerin | Adaptive Hypermedia | Ontologies | Semantic Web | Open Hypermedia Systems |
| 4 | HTML | Semantic Web | Intelligent Agent | Unix | Electronic Publishing | Open Hypermedia |
| 5 | Artificial Intelligence | Robotics | Intelligent Agent | Autonomous Agents | Hypermedia | Machine Learning |
| 6 | Multiagent System | Applied Artificial Intell | Robotics | Hypermedia | Hypertext | Software Engineering |
| 7 | Autonomous Agents | Semantic Web | Robotics | Software Engineering | Hypertext | Hypermedia |
| 8 | Hypermedia | Ontologies | XML | Artificial Intelligence | Adaptive Hypermedia | HTML |
| 9 | Hypertext | Adaptive Hypermedi | Intelligent Agent | XML | Artificial Intelligence | Semantic Web |
| 10 | Software Engineerin | Intelligent Agent | Simulation | Autonomous Agents | UML | Semantic Web |
| 11 | Hardware Design | Hardware | Semantic Web | Trusted Software Agents | Semantic Web Applicati | Preserving Temporal Logic |
| 12 | Pattern Recognition | Robotics | Classification | Agents | Texture Classification | Pattern Analysis |
| 13 | Machine Learning | Software Engineerin | Neural Information Pr | Natural Language | Robotics | Ontology |
| 14 | Signal Processing | Classification | Software Engineering | Digital Signal Processing | Machine Learning | Simulation |
| 15 | Logic Programming | Intelligent Agent | Robotics | Signal Processing | Image Processing | Multiagent System |
| 16 | Concurrency | Signal Processing | Semantic Web | Autonomous Agents | Robotics | Hypertext |
| 17 | Simulation | VLSI | Hypertext | Hypermedia | Autonomous Agents | Robotics |
| 18 | Ontologies | Software Engineerin | XML | Ontology | Knowledge Acquisition | Artificial Intelligence |
| 19 | Image Processing | Hypertext | Classification | Agents | Software Engineering | Robotics |
| 20 | Connectionism | Induction | Turing Testing | Building Situated Embodied | Agents | Theoretical Artificial Intellige |
| 21 | Robotics | Software Engineerin | Semantic Web | Image Processing | Machine Learning | Autonomous Agents |
| 22 | Semantic Web | Open Hypermedia | XML | Adaptive Hypermedia | Open Hypermedia | Knowledge Management |
| 23 | Information Retrieval | Knowledge Manage | Information Processin | Hypermedia With Dynamic L | Unix | Artificial Intelligence |
| 24 | User Interface | HCI | Machine Learning Re | Information Retrieval | Agents | Artificial Intelligence |
| 25 | Classification | Computer Vision | Agents | Simulation | Hypertext | Pattern Recognition |
| 26 | Fault Tolerance | Image Processing | Concurrency | Digital Signal Processing | Hardware | Signal Processing |
| 27 | VLSI Circuits | Co Design Methodol | Variable Length Input | Data Compression | Test Cost Reduction Th | Scheduling |
| 28 | Data Compression | Hypertext | VLSI Design Project | Image Processing | Signal Processing | Intelligent Agent |
| 29 | Computer Graphics | OLogic Programmin | Signal Processing | Data Compression | Machine Learning | Intelligent Agent |

FIGURE 5. Sample dataset used comprising 30 research areas.



| 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|
| Autonomous Agents | Artificial Intelligence | Semantic Web | Hypermedia | Multiagent System |
| Scheduling | Testing | Artificial Intelligence | Simulation | CAD |
| XML | UML B Specification | UML Models | UML B | Software Engineering |
| Open Hypermedia | Hypermedia | Agents | HTML | Hypertext |
| Open Hypermedia Systems | XML | Hypermedia | Hypertext | Agents |
| Semantic Web | Hypertext | Multiagent System | Agents | Software Engineering |
| Multi Agents | Intelligent Agent | Artificial Intelligence | Autonomous Agents | Agents |
| Intelligent Agent | Multi Agents | Artificial Intelligence | Agents | Multiagent System |
| Open Hypermedia | Open Hypermedia Systems | Semantic Web | Agents | Hypertext |
| HTML | Open Hypermedia | Agents | Open Hypermedia Systems | Hypermedia |
| Hypermedia | Hypertext | Multiagent System | Agents | Artificial Intelligence |
| Embedded System | UML Distilled | UML | Software Engineering | Concurrency |
| Machine Learning | Signal Processing | Artificial Intelligence | Image Processing | Computer Vision |
| Computer Vision | Agents | Intelligent Systems | Image Processing | Artificial Intelligence |
| Pattern Recognition | Computer Vision | Robotics | Image Processing | Artificial Intelligence |
| Agents | Hypertext | Logic Program Synthesis | Artificial Intelligence | Software Engineering |
| Agents | Hypermedia | Artificial Intelligence | Simulation | Software Engineering |
| Signal Processing | Multiagent System | Software Engineering | Artificial Intelligence | Agents |
| Hypertext | Knowledge Management | Hypermedia | Agents | Semantic Web |
| Signal Processing | Artificial Intelligence | Machine Learning | Pattern Recognition | Computer Vision |
| Natural Language | Experimental Artificial Intellig | Ontology | Machine Learning | Artificial Intelligence |
| Signal Processing | Multiagent System | Hypermedia | Agents | Artificial Intelligence |
| Artificial Intelligence | Hypertext | Ontologies | Agents | Open Hypermedia |
| Hypermedia Links | Agents | Open Hypermedia Systems | Hypermedia | Hypertext |
| Unix | HTML | Open Hypermedia Systems | Hypermedia | Hypertext |
| Hypermedia | Machine Learning | Artificial Intelligence | Signal Processing | Image Processing |
| Artificial Intelligence | VLSI | Simulation | VLSI Systems | Classification |
| BIST Hardware Synthesis | High level Simulation | CAD | Power Constrained Testing | Simulation |
| Computer Graphics | Advanced Computer Graphi | Artificial Intelligence | Computer Vision | Software Engineering |
| Hypertext | Software Engineering | Artificial Intelligence | Image Processing | Computer Vision |

FIGURE 6. Sample dataset – continued.

```
16 multivariate
16 splines
16 polynomial
17 regression
17 classification
17 information processing
17 learning
17 machine learning
18 data structures
18 algorithm analysis
19 parallelism
19 geometric
20 markup language
20 xml
20 html
21 interpreters
21 logic programming
21 deduction
21 program synthesis
22 reconstruction
22 wavelet
23 software
23 channels
23 coding
23 video
23 signal processing
23 complexity
23 neural networks
24* artificial intelligence
24* programming language
24* networks
24* distributed systems
25 image processing
25 pattern recognition
25 shape
25 computer vision
26 arrays
26 simulation
26 parallel processing
26 parallel
27 atm
27 ethernet
27 fddi
28 web services
28 owl
28 agents
28 rdf
28 semantic web
28 daml
28 frameworks
```

Figure 7. CORDER clusters for 51 of the 343 areas of the dataset. The clustering distance threshold was set to 0.5 (i.e. 50). Numbers represent the cluster computed. Observe that artificial intelligence is contained by cluster 24.